

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



УТВЕРЖДАЮ

Проректор по образовательной деятельности КФУ

Проф. Д. А. Таюрский

» \_\_\_\_\_ 20\_\_ г.

*подписано электронно-цифровой подписью*

## Программа дисциплины

Программирование на C#

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: 2020

## Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО
2. Место дисциплины (модуля) в структуре ОПОП ВО
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 (модуля)
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
6. Фонд оценочных средств по дисциплине (модулю)
7. Перечень литературы, необходимой для освоения дисциплины (модуля)
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 (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
13. Приложение №1. Фонд оценочных средств
14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)
15. Приложение №3. Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Программу дисциплины разработал(а)(и) заместитель директора по научной деятельности Тумаков Д.Н. (Директорат Института ВМ и ИТ, Институт вычислительной математики и информационных технологий), Dmitri.Tumakov@kpfu.ru

### 1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-3	Способен применять знания и методы дисциплин естественно-научного и математического цикла при проведении научных исследований, в том числе математического и компьютерного моделирования и высокопроизводительных вычислений
ПК-4	Разработка, отладка, рефакторинг программного кода, баз данных, информационных ресурсов; проектирование и интеграция программного обеспечения, управление проектами в области ИТ

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

принципы "классического" программирования

Должен уметь:

программировать на языке C#

Должен владеть:

основами языка C#

Должен демонстрировать способность и готовность:

практические навыки программирования и разработки начальных приложений в среде Visual Studio на языке C#

### 2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в раздел "Б1.В.ДВ.03.01 Дисциплины (модули)" основной профессиональной образовательной программы 01.04.02 "Прикладная математика и информатика (Математическое моделирование)" и относится к дисциплинам по выбору.

Осваивается на 1 курсе в 1 семестре.

### 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

Общая трудоемкость дисциплины составляет 2 зачетных(ые) единиц(ы) на 72 часа(ов).

Контактная работа - 18 часа(ов), в том числе лекции - 0 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 18 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 54 часа(ов).

Контроль (зачёт / экзамен) - 0 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 1 семестре.

### 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)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
	Тема 1. Обзор платформы MS.NET.					

Обзор языка C#. Использование структурных переменных. Операторы и исключения.

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 индексы. Атрибуты. Агрегации, пространства имен, сборки и модули.	1	0	0	4	14
	Итого		0	0	18	54

#### 4.2 Содержание дисциплины (модуля)

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

Общезыковая 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 отсоединенной 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 реализуют 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 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 своё состояние недоступным. Нам не интересно, 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 - тэг 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"

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

## **6. Фонд оценочных средств по дисциплине (модулю)**

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

В фонде оценочных средств содержится следующая информация:

- соответствие компетенций планируемым результатам обучения по дисциплине (модулю);
- критерии оценивания сформированности компетенций;
- механизм формирования оценки по дисциплине (модулю);
- описание порядка применения и процедуры оценивания для каждого оценочного средства;
- критерии оценивания для каждого оценочного средства;
- содержание оценочных средств, включая требования, предъявляемые к действиям обучающихся, демонстрируемым результатам, задания различных типов.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

## **7. Перечень литературы, необходимой для освоения дисциплины (модуля)**

Освоение дисциплины (модуля) предполагает изучение основной и дополнительной учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями;

- в печатном виде - в Научной библиотеке им. Н.И. Лобачевского. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования Научной библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,5 экземпляра (для обучающихся по ФГОС 3++ - не менее 0,25 экземпляра) каждого из изданий основной литературы и не менее 0,25 экземпляра дополнительной литературы на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину.

Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов Научной библиотеки КФУ.

### 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://znanium.com/catalog.php?bookinfo=204273>

Команда MS Teams "Программирование на C#" -

<https://teams.microsoft.com/l/team/19%3a183f8dd52a61496fb30971464469d517%40thread.tacv2/conversations?groupId=2a4d>

Методы программирования. Компьютерные вычисления - <http://znanium.com/bookread.php?book=350418>

Практикум по курсу - [http://libweb.ksu.ru/ebooks/09\\_63.pdf](http://libweb.ksu.ru/ebooks/09_63.pdf)

Руководство по программированию на C# - <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. Реализация данной дисциплины предполагает как очное, так и дистанционное форму обучения.
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.
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)

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

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

Материально-техническое обеспечение образовательного процесса по дисциплине (модулю) включает в себя следующие компоненты:



Помещения для самостоятельной работы обучающихся, укомплектованные специализированной мебелью (столы и стулья) и оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду КФУ.

Учебные аудитории для контактной работы с преподавателем, укомплектованные специализированной мебелью (столы и стулья).

Компьютер и принтер для распечатки раздаточных материалов.

Компьютерный класс.

## **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".

### Перечень литературы, необходимой для освоения дисциплины (модуля)

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: 2020

#### Основная литература:

1. Александрова, И.Л. Программирование на языке C#: учебное пособие. - 2-е издание / И.Л. Александрова, Д.Н. Тумаков. - Казань: Издательство Казанского университета, 2017. - 112 с. ISBN 978-5-00019-944-5 - Текст : электронный. - URL: [https://shelly.kpfu.ru/e-ksu/docs/F\\_2033489788/Tumakov\\_\\_\\_Programmirovanie\\_na\\_yazyke\\_C\\_.pdf](https://shelly.kpfu.ru/e-ksu/docs/F_2033489788/Tumakov___Programmirovanie_na_yazyke_C_.pdf) (дата обращения: 11.03.2020). - Режим доступа: открытый.
2. Гуриков, С. Р. Введение в программирование на языке Visual C# : учебное пособие / С.Р. Гуриков. - Москва : ФОРУМ : ИНФРА-М, 2020. - 447 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-00091-458-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1092167> (дата обращения: 11.03.2020). - Режим доступа: по подписке.
3. Кузин, А. В. Программирование на языке Си: учебное пособие / А.В. Кузин, Е.В. Чумакова.- Москва: ФОРУМ : ИНФРА-М, 2019. - 144 с. - (Высшее образование). - ISBN 978-5-00091-066-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1007488> (дата обращения: 11.03.2020). - Режим доступа: по подписке.

#### Дополнительная литература:

1. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# : учебное пособие / Л. А. Залогова. - 2-е изд., стер. - Санкт-Петербург : Лань, 2020. - 192 с. - ISBN 978-5-8114-4757-2. - Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/126160> (дата обращения: 11.03.2020). - Режим доступа: для авториз. пользователей.
2. Солдатенко, И. С. Практическое введение в язык программирования Си : учебное пособие / И. С. Солдатенко, И. В. Попов. - Санкт-Петербург : Лань, 2018. - 132 с. - ISBN 978-5-8114-3150-2. - Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/109619> (дата обращения: 11.03.2020). - Режим доступа: для авториз. пользователей.

Приложение 3  
к рабочей программе дисциплины (модуля)  
Б1.В.ДВ.03.01 Программирование на C#

**Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем**

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: 2020

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

Операционная система Microsoft Windows 7 Профессиональная или Windows XP (Volume License)

Пакет офисного программного обеспечения Microsoft Office 365 или Microsoft Office Professional plus 2010

Браузер Mozilla Firefox

Браузер Google Chrome

Adobe Reader XI или Adobe Acrobat Reader DC

Kaspersky Endpoint Security для Windows