

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  
Московский государственный университет имени М.В. Ломоносова  
филиал МГУ в г. Севастополе  
факультет компьютерной математики  
кафедра программирования

УТВЕРЖДАЮ  
Директор  
Филиала МГУ в г. Севастополе  
О.А. Шпырко  
«15» июня 2020 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**  
**Наименование дисциплины (модуля):**

*АЛГОРИТМЫ И АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ*

*код и наименование дисциплины (модуля)*

**Уровень высшего образования:**  
*бакалавриат*

**Направление подготовки:**

*01.03.02 Прикладная математика и информатика*

*(код и название направления/специальности)*

**Направленность (профиль) ОПОП:**  
*общий*

*(если дисциплина (модуль) относится к вариативной части программы)*

**Форма обучения**

*очная*

Рабочая программа рассмотрена на заседании кафедры программирования протокол № 3 от «28» июня 2020 г. Руководитель ОП 01.03.02 «Прикладная математика и информатика»  
Н. В. Лактионова  
(подпись)

Рабочая программа одобрена Методическим советом Филиала МГУ в г. Севастополе Протокол № 6 от «10» июня 2020 г.  
А.В. Мартынкин  
(подпись)

Севастополь, 2020

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по направлению подготовки 01.03.02 «Прикладная математика и информатика» в редакции приказа МГУ от 30 декабря 2016 г.

*курс – I*

*семестр – I*

*зачетных единиц – 4 кредита;*

*академических часов – 144, в т.ч.:*

*лекций – 54 часов*

*самостоятельной работы – 90 часов*

*практических занятий – нет*

*Формы промежуточной аттестации:*

*зачёты в семестрах – нет*

*Форма итоговой аттестации:*

*экзамен в I семестре.*

## ОГЛАВЛЕНИЕ

1. ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ .....	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП .....	3
3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....	3
4. СТРУКТУРА УЧЕБНОЙ ДИСЦИПЛИНЫ .....	5
4.3 СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИНЫ .....	7
5. РЕКОМЕНДУЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ .....	10
6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ .....	10
7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....	<b>ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.</b>
8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ .....	14
9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	14

## 1. ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

**Целью** освоения дисциплины является развитие контекста алгоритмического мышления, как культурной формы деятельности, определяемой как структурными особенностями алгоритмического знания, так и местом программирования в системе наук.

### **Задачи дисциплины:**

- создать представление о том, как возникали и развивались основные алгоритмические методы, понятия и идеи программирования;
- определить роль и место программирования в системе математических знаний;
- выяснить характер и особенности развития алгоритмического знания в определенные исторические периоды, оценить вклад, внесенный в программирование великими учеными;
- проанализировать исторический путь отдельных дисциплин программирования и выявить их связь с потребностями людей и задачами других наук;
- установить связи между различными разделами математики и программированием;
- овладеть навыками работы с литературой, особенностями библиографического поиска, научиться корректно цитировать и ссылаться на использованные материалы.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина относится к базовой части ОС МГУ. Содержание курса определяется образовательным стандартом МГУ высшего профессионального образования по направлению 01.03.02 Прикладная математика и информатика (1 ступень бакалавриата двухуровневой программы интегрированный магистр непрерывной подготовки).

Изучаются системы счисления, способы записи алгоритмов, машины Тьюринга и алгоритмы Маркова, устройство ЭВМ, системы команд ЭВМ, мнемокоды и алгоритмические языки, алгоритмический язык Паскаль, технологии программирования и отладки, сложность алгоритмов, структуры данных. Изучение опирается на школьную математику и информатику, а также на курсы алгебры и геометрии, математического анализа.

Основу данного курса составляют технологии алгоритмизации и программирования на языках высоко уровня, а также методы реализации базовых структур данных.

Освоение данной дисциплины (модуля) необходимо для дальнейшего изучения курсов «Операционные системы», «Языки программирования», «Генетические алгоритмы и нечеткая обработка данных», методов приближенных вычислений и большинства специальных курсов.

## 3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате освоения дисциплины обучающийся должен:

### **Знать:**

- неформальное и формальные определения понятия «алгоритм»;

- основные способы конструирования алгоритмов;
- определения эквивалентности машин Тьюринга;
- существование универсальной машины Тьюринга;
- существование алгоритмически неразрешимых проблем, методы доказательства алгоритмической неразрешимости;
- язык программирования C#(C++), структуру C#(C++)-программы;
- базовые алгоритмы решения задач сортировки, поиска, топологической сортировки, работы с текстами;
- основные структуры данных: стек, очередь, список, дерево и т.п..

***Уметь:***

- строить алгоритмы для решения простых задач в алгоритмических системах Тьюринга и Маркова;
- строить универсальную машину Тьюринга, доказывать алгоритмическую неразрешимость конкретных проблем;
- составлять и отлаживать программы на языке C#(C++);
- применять базовые алгоритмы и основные структуры данных, изучаемые в курсе, при разработке программ;
- оценивать сложность алгоритмов при их выборе.

***Владеть:***

- Современной технологией разработки и отладки программ на языке C#(C++).

## 4. СТРУКТУРА УЧЕБНОЙ ДИСЦИПЛИНЫ

### 4.1 Общая трудоемкость дисциплины:

- зачетных единиц – 4
- академических часов – 144 ч.
- лекций – 54 ч.
- самостоятельная работа – 90 ч.

### 4.2 Тематический план

Таблица 1.

### ТЕМАТИЧЕСКИЙ ПЛАН

#### Семестр I

№ п/п	Название темы	Количество часов			Формы текущего контроля успеваемости (по темам) / Форма промежуточной аттестации (по семестрам)
		Л	С(П,Л6)	СРС	
1	2	3	4	5	6
1.	Введение в теорию алгоритмов.	2		4	Консультации
2.	Машина Тьюринга.	2		4	Консультации
3.	Алгоритмически неразрешимые проблемы.	2		4	Консультации
4.	Модельная ЭВМ.	2		4	Консультации
5.	Оценка сложности алгоритмов.	2		4	Консультации
6.	Коллоквиум по пройденному материалу.	2		4	Проверка выполнения задания самостоятельной работы. Ответы на вопросы по пройденному материалу.
7.	Алгоритмические языки.	2		2	Консультации
8.	Объектная ориентация языка C#(C++)	2		2	Консультации
9.	Типы данных в языке C#(C++).	2		2	Консультации
10.	Операции и целочисленные выражения.	2		2	Консультации
11.	Выражения с операндами базовых типов.	2		2	Консультации
12.	Типы C#(C++) и типы платформы .NET Framework	2		2	Консультации
13.	Операторы C#(C++)	2		2	Консультации
14.	Массивы C#(C++)	2		2	
15.	Строки — объекты класса string	2		2	

16.	Методы C#(C++)	2		4	
17.	Класс как совокупность статических членов	2		4	
18.	Коллоквиум по пройденному материалу.	2		4	Проверка выполнения задания самостоятельной работы. Ответы на вопросы по пройденному материалу.
19.	Классы как типы.	2		4	Консультации
20.	Средства взаимодействия с объектами.	2		4	Консультации
21.	Включение, вложение и наследование классов.	2		4	Консультации
22.	Интерфейсы.	2		4	Консультации
23.	Перечисления и структуры.	2		4	Консультации
24.	Исключения	2		4	Консультации
25.	Делегаты и события.	2		4	Консультации
26.	Обобщения.	2		4	Консультации
27.	Коллоквиум по пройденному материалу.	2		4	Проверка выполнения задания самостоятельной работы. Ответы на вопросы по пройденному материалу.
<b>Всего, часов</b>		<b>54</b>		<b>90</b>	
<b>Итоговый контроль</b>					<b>Экзамен</b>

где: Л – лекции, С – семинарские занятия, П – практические занятия, Лб – лабораторные занятия, СРС – самостоятельная работа студентов.

## 4.3 Содержание разделов дисциплины

### Планы лекций

#### **Лекция 1. Введение в теорию алгоритмов.**

Интуитивное понятие алгоритма. Свойства алгоритмов. Уточнение понятия алгоритма, алгоритм как преобразование слов из заданного алфавита.

#### **Лекция 2. Машина Тьюринга.**

Машина Тьюринга. Нормальные алгоритмы Маркова. Основные схемы объединения алгоритмов (композиция, разветвление, цикл). Тезис Тьюринга и принцип нормализации, их обоснование.

#### **Лекция 3. Алгоритмически неразрешимые проблемы.**

Алгоритмически неразрешимые проблемы. Неразрешимость проблем самоприменимости, останова и эквивалентности алгоритмов.

#### **Лекция 4. Модельная ЭВМ.**

Модельная ЭВМ. Машинное представление данных. Машинный язык как практический формальный способ записи алгоритмов. Недостатки машинных языков.

#### **Лекция 5. Оценка сложности алгоритмов.**

Оценка сложности алгоритмов. Понятие о временной оценке сложности алгоритмов в худшем случае и в среднем.

#### **Лекция 6. Алгоритмические языки.**

Характеристика алгоритмических языков высокого уровня. Понятие трансляции. Алфавит, синтаксис, семантика и прагматика алгоритмического языка. Описание синтаксиса языка с помощью металингвистических формул (БНФ) и синтаксических диаграмм.

#### **Лекция 7. Объектная ориентация языка C#(C++).**

Типы, классы, объекты. Программа на C#. Пространство имен. Создание консольного приложения.

#### **Лекция 8. Типы данных в языке C#(C++).**

Типы ссылок и типы значений. Классификация типов C#. Простые (базовые) типы. Константы-литералы. Объявления переменных и констант базовых типов.

#### **Лекция 9. Операции и целочисленные выражения.**

Операции языка C#. Операции присваивания и оператор присваивания. Операции инкремента (++) и декремента (--). Выражения с арифметическими операциями. Поразрядные (побитовые) операции. Переполнения при операциях с целыми.

#### **Лекция 10. Выражения с операндами базовых типов.**

Автоматическое и явное приведение арифметических типов. Особые ситуации в арифметических выражениях. Логический тип и логические выражения. Выражения с символьными операндами. Тернарная (условная) операция.

#### **Лекция 11. Типы C#(C++) и типы платформы .NET Framework.**

Платформа .NET Framework и спецификация CTS. Простые (базовые) типы C# как типы CTS. Специфические методы и поля простых типов.

**Лекция 12. Операторы C#(C++).**

Общие сведения об операторах. Метки и оператор безусловного перехода. Условный оператор. Операторы цикла. Операторы передачи управления. Переключатель.

**Лекция 13. Массивы C#(C++).**

Одномерные массивы. Массивы как наследники класса Array. Виды массивов и массивы многомерные. Массивы массивов и «непрямоугольные» массивы. Массивы массивов и поверхностное копирование.

**Лекция 14. Строки — объекты класса string.**

Строковые литералы. Строковые объекты и ссылки типа string. Операции над строками. Некоторые методы и свойства класса string. Форматирование. Форматирование в ToString() и интерполяция строк. Применение строк в переключателях. Массивы строк. Сравнение строк. Преобразования с участием строкового типа. Аргументы метода Main(). Неизменяемость объектов класса string.

**Лекция 15. Методы C#(C++).**

Методы-процедуры и методы-функции. Методы локальные и сжатые до выражений. Соотношение параметров и аргументов. Параметры с типами ссылок. Методы с переменным числом аргументов. Перегрузка методов. Рекурсивные методы. Применение метода Array.Sort(). Кортежи и ссылки в методах.

**Лекция 16. Класс как совокупность статических членов.**

Статические члены класса. Поля классов (статические поля). Статические константы. Статические методы. Статический конструктор и статический класс.

**Лекция 17. Классы как типы.**

Объявление класса. Поля объектов. Объявления методов объектов. Пример класса и его объектов. Ссылка this. Конструкторы объектов класса. Деструкторы и инициализаторы объектов.

**Лекция 18. Средства взаимодействия с объектами.**

Принцип инкапсуляции и свойства классов. Автореализуемые свойства и свойства, сжатые до выражений. Индексаторы. Расширяющие методы и деконструкторы.

**Лекция 19. Включение, вложение и наследование классов.**

Включение объектов классов. Вложение классов. Наследование классов. Доступность членов класса при наследовании. Методы при наследовании. Абстрактные методы и абстрактные классы. Опечатанные классы и члены классов. Применение абстрактных классов.

**Лекция 20. Интерфейсы.**

Два вида наследования в ООП. Объявления интерфейсов. Реализация интерфейсов. Интерфейс как тип. Интерфейсы и наследование.

**Лекция 21. Перечисления и структуры.**

Перечисления. Базовый класс перечислений. Структуры. Упаковка и распаковка. Реализация структурами интерфейсов.

**Лекция 22. Исключения.**

О механизме исключений. Системные исключения и их обработка. Свойства исключений. Исключения в арифметических выражениях. Генерация исключений. Пользовательские классы исключений.

**Лекция 23. Делегаты и события.**

Синтаксис делегатов. Массивы делегатов. Многоадресные экземпляры делегатов. Делегаты и обратные вызовы. Анонимные методы и лямбда-выражения. События.

**Лекция 24. Обобщения.**

Заключительные выводы. Основные направления развития языка C#. Особенности применения в различных проектах.

## 5. РЕКОМЕНДУЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Учебный процесс при преподавании курса основывается на использовании традиционных, инновационных и информационных образовательных технологий. Традиционные образовательные технологии представлены лекциями. Инновационные образовательные технологии используются в виде широкого применения активных и интерактивных форм проведения занятий. Информационные образовательные технологии реализуются путем активизации самостоятельной работы студентов в информационной образовательной среде.

Работа в аудитории: лекции; консультации перед экзаменом.

Процесс изложения учебного материала сопровождается демонстрацией решения задач в интерактивном режиме.

Внеаудиторная работа: изучение пройденных на лекциях тем, самостоятельное изучение литературы по дисциплине, выполнение самостоятельных заданий, направленных на освоение материала и программную реализацию рассмотренных методов.

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная работа студентов – это процесс активного, целенаправленного приобретения студентом новых знаний, умений без непосредственного участия преподавателя, характеризующийся предметной направленностью, эффективным контролем и оценкой результатов деятельности обучающегося.

Цели самостоятельной работы:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную и справочную документацию, специальную литературу;
- развитие познавательных способностей, активности студентов, ответственности и организованности;
- формирование самостоятельности мышления, творческой инициативы, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений и академических навыков.

Самостоятельная работа может осуществляться индивидуально или группами студентов в зависимости от цели, объема, уровня сложности, конкретной тематики.

Технология организации самостоятельной работы студентов включает использование информационных и материально-технических ресурсов образовательного учреждения.

Перед выполнением обучающимися внеаудиторной самостоятельной работы преподаватель может проводить инструктаж по выполнению задания. В инструктаж включается:

- цель и содержание задания;
- сроки выполнения;
- ориентировочный объем работы;
- основные требования к результатам работы и критерии оценки;
- возможные типичные ошибки при выполнении.

Инструктаж проводится преподавателем за счет объема времени, отведенного на изучение дисциплины.

Контроль результатов внеаудиторной самостоятельной работы студентов может проходить в письменной, устной или смешанной форме.

Студенты должны подходить к самостоятельной работе как к наиважнейшему средству закрепления и развития теоретических знаний, выработке единства взглядов на отдельные вопросы курса, приобретения определенных навыков и использования профессиональной литературы.

Помещения для самостоятельной работы обучающихся должны быть оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

При самостоятельной проработке курса обучающиеся должны:

- просматривать основные определения и факты;
- повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;
- изучить рекомендованную литературу, составлять тезисы, аннотации и конспекты наиболее важных моментов;
- самостоятельно выполнять задания, аналогичные предлагаемым на занятиях;
- использовать для самопроверки материалы фонда оценочных средств;
- выполнять домашние задания по указанию преподавателя.

**Самостоятельная работа по изучению данной дисциплины включает:**

- проработку теоретических основ лекционного материала;
- систематизацию изученного материала по курсу;
- научно-исследовательская работа учащегося в библиотеках;
- выполнение самостоятельных заданий, направленных на углубление знаний материала и программную реализацию рассмотренных методов, в соответствии с методическими указаниями.

Отчет по выполненным самостоятельным заданиям предполагает демонстрацию разработанных программ, реализующих изучаемые методы, и защиту выполненных заданий в ходе устного опроса преподавателя.

**Тематика самостоятельных заданий:**

Задание №1. Простые вычисления по формулам.

Задание №2. Накопление сумм и произведений в цикле.

Задание №3. Вычисление корня функционального уравнения.

Задание №4. Табулирование функции с выводом результата по логическому условию.

Задание №5. Логическая проверка попадания точки в заданную область.

Задание №6. Сложное условие завершения цикла.

Задание №6. Обработка одномерных массивов с единственным циклом.

Задание №6. Сложное условие завершения цикла при работе с матрицами.

Задание №6. Обработка матриц произвольного размера.

По итогам освоения дисциплины проводится **устный экзамен.**

## 7. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

### Рейтинг-план дисциплины.

#### 7.1. Соотношение видов учебной деятельности студента, учитываемых в рейтинговой оценке по данной дисциплине

№	Виды учебной деятельности студентов	Весовые коэффициенты, %
1	<i>Лекции</i>	<i>10</i>
2	<i>Самостоятельные работы</i>	<i>30</i>
3	<i>Коллоквиумы</i>	<i>60</i>
	Сумма	100%

Максимально возможные баллы за виды контролируемой учебной деятельности студента (оценки «идеального студента» за одну единицу учебной работы)

№	Содержание вида контролируемой учебной деятельности	Единица измерения работы	Максимальное количество баллов за единицу выполняемой работы
1.1	<i>Посещение одной лекции</i>	<i>1 задание</i>	<i>2</i>
1.2	<i>Самостоятельная работа</i>	<i>1 задание</i>	<i>15</i>
1.3	<i>Коллоквиум</i>	<i>1 задание</i>	<i>35</i>

#### 7.2. Список вопросов к устному экзамену

1. Задачи обработки информации и алгоритмы. Неформальное (интуитивное) определение алгоритма.
2. Формализация алгоритма. Машина Тьюринга.
3. Способы представления машин Тьюринга. Нормальные вычисления.
4. Диаграммы Тьюринга. Построение диаграмм Тьюринга. Построение таблиц по диаграммам.
5. Понятие универсальной машины Тьюринга. Построение универсальной машины Тьюринга.
6. Проблема останова и алгоритмическая неразрешимость.
7. Алгоритмическая неразрешимость проблемы самоприменимости.
8. Тезис Тьюринга - Чёрча.
9. Нормальные алгоритмы Маркова. Эквивалентность формальных систем описания алгоритмов.
10. Среда NET. Основные понятия (открытость, единый каркас среды, модульность, примитивные и структурные типы)
11. Общеязыковая интеграция в NET. Дизассемблирование. Метаданные
12. Пространство имен System. Структуры, определенные в пространстве System.
13. Класс Console/ Основные свойства и методы.
14. Система типов языка C#
15. Наследование в языке C#
16. Понятие класса в языке C# Поля, статические поля, свойства.
17. Виртуальные и перегруженные методы.
18. Понятие модуля в языке C#. Примеры
19. Понятие интерфейса в языке C# . Примеры

20. Массивы в C#. Класс Array
21. Строки в C#. Класс String
22. Коллекции. Основные понятия Типы Коллекций
23. Необобщенные коллекции. Основные классы. Примеры
24. Обобщенные коллекции. Основные классы. Примеры
25. Понятие делегата. Типы делегатов. Примеры использования одиночного делегата
26. Комбинированные делегаты. Определения. Примеры.
27. Функции обратного вызова
28. Событийная модель на основе делегатов
29. Особенности использования событий
30. Атрибуты в C#. определения. Предопределенные атрибуты в пространстве System
31. Создание собственных атрибутов в приложении
32. Рефлексия. Класс Reflection/примеры применения
33. Организация системы ввода-вывода в C#. Байтовый поток.
34. Символьный поток.
35. Двоичные потоки ввода-вывода
36. Перенаправление стандартных потоков. Примеры .
37. Оконные приложения в C#. Структура. Основные методы класса Form
38. Общая характеристика элементов управления. Класс Control
39. Программирование с использованием атрибутов
40. Основные понятия сетевого программирования
41. Стек протоколов TCP/IP
42. Структура клиент-серверного приложения на основе протокола TCP.
43. Структура клиент-серверного приложения на основе протокола UDP.
44. Структура многопоточного TCP сервера на основе протокола TCP.
45. Понятие потока. Диаграмма состояний потока.
46. Пул потоков.
47. Фоновые потоки
48. Синхронизация потоков с помощью класса Monitor, оператора lock
49. Синхронизация потоков с помощью классов AvtoResetEvent, Mutex.

#### **Критерии оценивания знаний студента на экзамене.**

**Отлично.** Обучающийся в полной мере владеет теоретическими основами дисциплины и научной терминологией, грамотно излагает материал и способен иллюстрировать ответ примерами, фактами, данными научных исследований, применять теоретические знания для решения практических задач.

**Хорошо.** Обучающийся владеет теоретическими основами дисциплины и научной терминологией, грамотно излагает материал и способен иллюстрировать ответ примерами, фактами, данными научных исследований, применять теоретические знания для решения практических задач, но допускает отдельные несущественные ошибки.

**Удовлетворительно.** Обучающийся владеет частично теоретическими основами дисциплины и научной терминологией, фрагментарно способен иллюстрировать ответ примерами, допускает несколько существенных ошибок в ответе..

**Неудовлетворительно.** Обучающийся не владеет теоретическими основами дисциплины и научной терминологией, демонстрирует отрывочные знания, не способен иллюстрировать ответ примерами, допускает множественные существенные ошибки в ответе..

## 8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 8.1 Основная литература

1. Подбельский В. В. Программирование. Базовый курс C# : учебник для бакалавриата и специалитета / В. В. Подбельский. — Москва : Издательство Юрайт, 2019. — 369 с. — (Бакалавр и специалист). — ISBN 978-5-534-10616-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/439068> (дата обращения: 24.08.2019).
2. Крупский В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/444131> (дата обращения: 25.08.2019).

### 8.2. Дополнительная литература

1. Дискретная математика: прикладные задачи и сложность алгоритмов : учебник и практикум для академического бакалавриата / А. Е. Андреев, А. А. Болотов, К. В. Коляда, А. Б. Фролов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 317 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-04246-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/444120> (дата обращения: 25.08.2019).

## 9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

- библиотека Филиала МГУ в г. Севастополе;
- библиотека кафедры программирования Филиала МГУ в г. Севастополе;
- лекционная аудитория, оборудованную средствами подключения к сети электропитания и локальной сети университета, а также средствами интерактивного отображения информации для показа презентаций лекций и демонстрации решения задач.
- для самостоятельной работы студентов - специализированные компьютерные классы с доступом к Интернет-ресурсам с любого компьютера.

**Пример экзаменационного билета****ФИЛИАЛ МОСКОВСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА  
имени М.В.ЛОМОНОСОВА в г. СЕВАСТОПОЛЕ**

Направление подготовки 01.03.02 «Прикладная математика и информатика»  
Учебная дисциплина Алгоритмы и алгоритмические языки  
Семестр 1

**Экзаменационный билет № 1**

- 1) Формализация алгоритма. Машина Тьюринга..
- 2) Строки в C#. Класс String.
- 3) Разработать класс, набор методов (конструктор и минимум два метода) для программной модели заданного объекта (по вариантам). Описание объекта и его основных свойств приводится ниже. Привести фрагмент программы (static main), использующей объекты разработанного класса.

Утверждено на заседании кафедры прикладной математики (программирования)  
Протокол № \_\_\_ от «\_\_\_» \_\_\_\_\_ 2019 г.

Зав. кафедрой \_\_\_\_\_ С. И. Гуров  
Преподаватель \_\_\_\_\_ И.Ю. Гришин