


Министерство науки и высшего образования Российской Федерации  
ФГБОУ ВО «Дагестанский государственный технический университет»

РЕКОМЕНДОВАНО  
К УТВЕРЖДЕНИЮ  
Декан, председатель совета  
факультета КТВТиЭ,  
 Юсуфов Ш.А.

УТВЕРЖДАЮ  
Проректор по учебной работе,  
председатель методического  
совета ДГТУ  
 Суракатов Н.С.

«24» 09 2018г.

«26» 09 2018г.

РАБОЧАЯ ПРОГРАММА

Дисциплина Б1.В.ДВ.10 Объектно-ориентированное программирование  
для направления 01.03.02 Прикладная математика и информатика  
по профилю Системное программирование и компьютерные технологии  
факультет Компьютерных технологий, вычислительной техники и энергетики  
наименование факультета, где ведется дисциплина  
кафедра Программное обеспечение вычислительной техники и  
автоматизированных систем  
наименование кафедры, за которой закреплена дисциплина

Квалификация выпускника (степень) бакалавр  
Форма обучения очная; курс 4; семестр(ы) 8;  
Всего трудоемкость в зачетных единицах (часах) 2 зет(72);  
Лекции 16 (час); Экзамен - (сем);  
Практические (семинарские) занятия 16 (час); Зачет 8 (семестр);  
Лабораторные занятия - (час);  
Курсовой проект (работа, РГР) - (семестр);  
Самостоятельная работа 40 (час).

Зав. кафедрой  /Мелехин В.Б./  
подпись ФИО

Начальник УО  /Магомаева Э.В./  
подпись ФИО



Программа составлена в соответствии с требованиями ФГОС ВО с учетом рекомендаций ООП ВО по направлению 01.03.02 «Прикладная математика и информатика» и профилю подготовки «Системное программирование и компьютерные технологии».

Программа одобрена на заседании выпускающей кафедры от 16.09.2018 года, протокол № 1.  
Зав. выпускающей кафедрой по данному направлению (специальности,

профильно)  / Исабекова Т.И./

### ОДОБРЕНО

Методической комиссией по УГС и  
направления подготовки

01.00.00 «Математика и  
механика»

номер и полное наименование направления

Председатель МК

 /Исабекова Т.И./  
подпись, ФИО

«23» 09 2018г.

### АВТОР ПРОГРАММЫ

Камилова А.М.,

ФИО, уч. степень, ученое звание, пол-  
ность

ст. преподаватель



## **1. Цели освоения дисциплины**

Учебная дисциплина «Объектно-ориентированное программирование» реализует требования федерального государственного образовательного стандарта высшего образования по направлению бакалавриата 01.03.02 «Прикладная математика и информатика» и профилю подготовки «Системное программирование и компьютерные технологии».

Цель дисциплины – формирование у студентов теоретических знаний и практических навыков в области объектно-ориентированного программирования.

Задача дисциплины - привить обучаемым навыки использования методов объектно-ориентированного программирования при разработке программных систем разного уровня сложности.

## **2. Место дисциплины в структуре ООП бакалавриата**

«Объектно-ориентированное программирование» входит в цикл дисциплин по выбору (вариативная часть).

Для успешного усвоения данной дисциплины необходимо, чтобы обучаемый владел знаниями, умениями и навыками, сформированными в процессе изучения дисциплин “Основы информатики”, “Языки и методы программирования”, “Информационные системы и технологии”, “Системы программирования”.

## **3. Компетенции обучающегося, формируемые в результате освоения дисциплины «Объектно-ориентированное программирование»**

Процесс изучения дисциплины направлен на формирование у обучающегося следующих компетенций:

- способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям (ОПК-3);
- способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-4).

- способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения (ПК-7).

В результате изучения дисциплины обучаемый должен:

**знать:**

- сущность объектно-ориентированного программирования;
- развитие языков как развитие абстрактных моделей;
- элементы объектной модели;
- основные понятия объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм, класс как основной механизм абстракции, наследование как форма отношения обобщения, общий полиморфизм и виртуальные члены-функции;

**уметь:**

- разрабатывать программы задач информационных систем методом объектно-ориентированного программирования на Microsoft Visual C++;
- проводить тестирование и отладку программ;

**владеть:**

- навыками компонентного программирования.

#### 4. Структура и содержание дисциплины «Объектно-ориентированное программирование»

Общая трудоемкость дисциплины составляет **2 зачетные единицы – 72 часа**, в том числе – лекционных **16 часов**, практических **16 часов**, СРС **40 часов**, форма отчетности: 8 семестр – зачет.

##### 4.1. Содержание дисциплины

№ п/п	Раздел дисциплины Тема лекции и вопросы	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля (по срокам текущей аттестации)
				ЛК	ПЗ	ЛР	СРС	
1	2	3	4	5	6	7	8	9
1.	<p><b>Лекция 1.</b></p> <p><b>Тема: «Развитие языков как развитие абстрактных моделей».</b></p> <p>Абстрактные модели, лежащие в основе языков программирования. Абстрагирование в языках программирования. Причины появления и актуальность объектно-ориентированного подхода к программированию. Сущность ООП. Преимущества и недостатки ООП. Области применения ООП. Основные принципы ООП. Объектно-ориентированные языки программирования.</p>	8	1	2	2		2	Входной контроль
2.	<p><b>Лекция 2.</b></p> <p><b>Тема: «Понятие класса и объекта в С++».</b></p> <p>Назначение и семантика классов. Классы и абстрактные типы данных. Защита данных на уровне класса (модификаторы доступа). Объекты ООП и объекты реального мира. Представление данных. Классификация методов. Инкапсуляция данных.</p>	8	2	2	2		4	

3.	<p><b>Лекция 3.</b>  <b>Тема: «Конструкторы и деструкторы в C++».</b>  Создание объектов и массивов объектов. Правила для конструкторов и деструкторов в C++. Стандартный конструктор, конструктор с параметрами. Список инициализации. Конструктор копирования. Указатель this. Конструктор копирования. Глубинное и поверхностное копирование. Встраиваемые функции (inline).</p>	8	3	2	2		8	
4.	<p><b>Лекция 4.</b>  <b>Тема: «Члены класса, общие для всех экземпляров. Константные данные класса в C++».</b>  Назначение членов класса, общих для всех объектов класса. Статические переменные и статические функции класса (static). Константные переменные и функции класса (const). Функции, дружественные одному классу, а также нескольким классам. Методы класса, дружественные другому классу. Дружественные классы.</p>	8	4	2	2		4	
5.	<p><b>Лекция 5.</b>  <b>Тема: «Механизмы наследования в C++».</b>  Простое наследование. Спецификаторы доступа при наследовании. Конструкторы и деструкторы производных классов (очередность вызова). Переопределение и вызов элементов базового класса. Множественное наследование. Виртуальные базовые классы.</p>	8	5	2	2		6	Аттестационная контр. работа №1
6.	<p><b>Лекция 6.</b>  <b>Тема: «Перегрузка в C++».</b>  Назначение перегрузки. Перегрузка</p>	8	6	2	2		6	

	<p>функций. Декодирование компилятором имен функций. Перегрузка конструкторов. Неоднозначность, возникающая при перегрузке. Перегрузка операторов. Правила перегрузки операторов. Перегрузка унарных и бинарных операторов.</p>								
<b>7.</b>	<p><b>Лекция 7.</b> <b>Тема: «Полиморфизм и виртуальные функции в C++».</b> Виды и реализации полиморфизма. Механизмы раннего и позднего связывания. Динамический полиморфизм. Виртуальные функции. Виртуальные деструкторы. Чисто виртуальные функции и абстрактные классы.</p>	8	7	2	2		6		
<b>8.</b>	<p><b>Лекция 8.</b> <b>Тема: «Параметризованные типы в C++. Стандартная библиотека шаблонов C++».</b> Шаблоны функций. Специализация шаблонов функций. Шаблоны классов. Специализация шаблонов класса. Стандартная библиотека шаблонов STL. Перегрузка шаблонов. Контейнеры и алгоритмы библиотеки STL. Назначение и состав библиотеки. Контейнеры: последовательные и ассоциативные.</p>	8	8	2	2		4		
Итого						16	16	40	зачет

#### 4.2. Содержание практических занятий

№ п/п	№ лекции из рабочей программы	Наименование практического занятия	Рекомендуемая литература и методические разработки (№ источника из списка литературы)	Кол-во часов
1	Лк №1	Развитие языков как развитие абстрактных моделей. ООП как направление развития теории и практики создания ПО.	№1-16	2
2	Лк №2	Понятие класса и объекта в C++.	№1-16	2
3	Лк №3	Конструкторы и деструкторы в C++.	№1-16	2
4	Лк №4	Члены класса, общие для всех экземпляров. Константные данные класса в C++. Друзья класса (friend) в C++.	№1-16	2
5	Лк №5	Механизмы наследования в C++. Простое и множественное наследование.	№1-16	2
6	Лк №6	Перегрузка в C++. Перегрузка операторов.	№1-16	2
7	Лк №7	Полиморфизм и виртуальные функции в C++.	№1-16	2
8	Лк №8	Параметризованные типы в C++. Стандартная библиотека шаблонов C++.	№1-16	2
Итого				16

#### 4.3. Тематика для самостоятельной работы студента

№ п/п	Тематика по содержанию дисциплины, выделенная для самостоятельного изучения	Кол-во часов из содержания дисциплины	Рекомендуемая литература и источники информации	Форма контроля СРС
1.	Развитие языков как развитие абстрактных моделей. ООП как направление развития теории и практики создания ПО.	2	№1-16	Вх.контр



2.	Понятие класса и объекта в C++.	4	№1-16	
3.	Конструкторы и деструкторы в C++.	8	№1-16	
4.	Члены класса, общие для всех экземпляров. Константные данные класса в C++. Друзья класса (friend) в C++.	4	№1-16	
5.	Механизмы наследования в C++. Простое и множественное наследование.	6	№1-16	Аттестационная контр. работа №1
6.	Перегрузка в C++. Перегрузка операторов.	6	№1-16	зачет
7.	Полиморфизм и виртуальные функции в C++.	6	№1-16	зачет
8.	Параметризованные типы в C++. Стандартная библиотека шаблонов C++.	4	№1-16	зачет
Итого		40		

## **5. Образовательные технологии.**

В соответствии с требованиями ФГОС ВО по направлению подготовки реализация компетентностного подхода должна предусматривать широкое использование в учебном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Удельный вес занятий, проводимых в интерактивных формах, определяется главной целью программы, особенностью контингента обучающихся и содержанием дисциплины, и в целом в учебном процессе они составляют не менее 20% аудиторных занятий.

При проведении занятий по учебной дисциплине рекомендуется следовать и традиционным технологиям, в частности, в каждом разделе курса выделять наиболее важные моменты, акцентировать на них внимание обучаемых.

При чтении лекций по всем разделам программы иллюстрировать теоретический материал большим количеством примеров, что позволит сделать изложение наглядным и продемонстрировать обучаемым приемы программирования.

При изучении всех разделов программы добиться точного знания обучаемыми основных исходных понятий.

## **6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов**

### **Вопросы входного контроля для проверки знаний студентов**

1. Формат программы на C++.
2. Директивы препроцессора.
3. Основные типы данных C++.
4. Объявление переменных и констант в C++.
5. Операции C++.
6. Управляющие конструкции C++.
7. Объявление функций в C++. Способы передачи параметров функциям.
8. Составные типы данных в C++: массивы, структуры и объединения.
9. Указатели в C++. Работа с массивами и структурами с помощью указателей.
10. Динамическое распределение памяти.

### **Контрольные вопросы для проверки текущих знаний студентов**

#### **Аттестационная контрольная работа №1**

1. Анализ причин сложности современных программных систем.
2. Основы структурного подхода к программированию. Модульное программирование.
3. Абстрактные модели, лежащие в основе языков программирования.
4. Абстрагирование в языках программирования.
5. Причины появления и актуальность объектно-ориентированного подхода к программированию.
6. Сущность объектно-ориентированного подхода. Преимущества и недостатки объектного подхода.
7. Основные области применения ООП.
8. Преимущества инкапсуляции.
9. Понятие класса в ООП.
10. Состояние и поведение объекта.
11. Понятие объекта в ООП.
12. Суть наследования и его виды.
13. Полиморфизм.
14. Основные объектно-ориентированные языки программирования.
15. Структура класса C++. Уровни доступа к членам класса.

- 16.Объявление объектов. Создание массивов объектов.
- 17.Указатели на объекты класса. Динамические объекты и массивы объектов.
- 18.Синтаксис и семантика конструктора. Свойства конструкторов.
- 19.Конструктор по умолчанию. Конструктор с параметрами.
- 20.Конструктор со списком инициализаторов.
- 21.Конструктор копирования.
- 22.Синтаксис и семантика деструктора. Свойства деструкторов.
- 23.Указатель this. Встраиваемые функции.
- 24.Статические члены класса. Константные члены класса.
- 25.Модифицируемые члены класса.
- 26.Дружественные функции.
- 27.Дружественные классы.
- 28.Объявление классов при простом наследовании. Спецификаторы наследуемого доступа.
- 29.Конструкторы и деструкторы производных классов.
- 30.Виртуальные базовые классы.

### **Вопросы к зачету.**

1. Абстрактные модели, лежащие в основе языков программирования.
2. Абстрагирование в языках программирования.
3. История появления объектно-ориентированного программирования.
4. Основные принципы объектно-ориентированного программирования (перечислить, дать определение). Инкапсуляция. Понятие класса и объекта.
5. Основные принципы объектно-ориентированного программирования (перечислить, дать определение). Наследование и полиморфизм. Общие сведения.
6. Структура класса. Общий формат объявления класса в C++. Реализация методов класса. Модификаторы доступа private, public, protected. Пример объявления и реализации конкретного класса.
7. Создание объекта от класса в C++. Динамические объекты. Обращение к элементам класса через объекты (синтаксис). Пример. Ограничения на элементы класса. Неполное объявление класса в C++.
8. Конструктор и деструктор класса в C++. Правила объявления и вызова. Пример класса с конструктором и деструктором.
9. Стандартные конструктор и деструктор в C++. Пример программы.
- 10.Конструктор копирования в C++. Поверхностное и глубинное копирование данных. Примеры программ.

11. Конструктор с параметрами в C++. Пример объявления и реализации класса с конструктором с параметрами. Использование значения параметров по умолчанию. Пример программы.
12. Конструктор с параметрами в C++. Список инициализации. Пример использования списка инициализации.
13. Указатель **this** в C++. Назначение и пример использования.
14. Встраиваемые (**inline**) функции в C++. Требования, предъявляемые к **inline**-функциям. Пример программы.
15. Статические элементы класса в C++ (модификатор **static**). Статические переменные и функции класса. Пример использования статических элементов класса в программе.
16. Константные элементы класса в C++ (модификатор **const**). Константные объекты в C++. Примеры использования константных элементов в программе.
17. Дружественные функции в C++ (ключевое слово **friend**). Функции, дружественные одному классу. Пример их использования в программе.
18. Дружественные функции в C++ (ключевое слово **friend**). Функции, дружественные нескольким классам. Пример их использования в программе.
19. Дружественные функции в C++ (ключевое слово **friend**). Методы класса, дружественные другому классу. Пример их использования в программе.
20. Дружественные классы в C++. Правила, предъявляемые к дружественным классам. Пример их объявления в программе.
21. Принцип наследования в объектно-ориентированном программировании. Простое наследование в C++. Общий формат объявления производного класса в C++. Варианты наследуемого доступа к элементам базовых классов. Пример программной реализации простейшей иерархии классов.
22. Вызов конструкторов и деструкторов при наследовании в C++. Использование конструктора с параметрами при наследовании.
23. Переопределение и вызов элементов базового класса в производном классе. Пример программы.
24. Множественное наследование в C++. Общий формат объявления классов при множественном наследовании. Решение проблемы совпадения идентификаторов. Пример программы.
25. Вызов конструкторов и деструкторов при множественном наследовании в C++. Пример программы.
26. Виртуальные базовые классы в C++. Пример программы.
27. Перегрузка функции в C++. Перегрузка конструкторов. Неоднозначность, возникаемая при перегрузке методов класса. Пример программы.

28. Перегрузка операторов в C++. Правила перегрузки операторов. Перегрузка унарных операторов. Пример программы.
29. Перегрузка операторов в C++. Правила перегрузки операторов. Перегрузка бинарных операторов. Пример программы.
30. Механизм раннего и позднего связывания. Виртуальные функции в C++. Пример программы.
31. Виртуальные деструкторы в C++. Пример программы.
32. Абстрактные классы и чисто виртуальные функции в C++. Пример программы.
33. Шаблоны функций. Параметры шаблонов. Конкретизация шаблонной функции. Перегрузка шаблонов функции
34. Шаблоны классов. Статические члены шаблонных классов. Специализация шаблонов классов
35. Назначение и состав библиотеки STL. Контейнерные классы. Типы контейнеров.
36. Последовательные контейнеры: векторы, списки, деки. Адаптеры контейнеров.
37. Ассоциативные контейнеры: карты, мультикарты, множества, битовые множества.
38. Итераторы. Разновидности итераторов. Обратные итераторы
39. Алгоритмы. Модифицирующие и немодифицирующие операции с последовательностями.

### **Вопросы для контроля остаточных знаний**

1. Преимущества применения ООП при создании программного обеспечения.
2. Основные концепции ООП.
3. Понятие объекта.
4. Понятие класса.
5. Модификаторы доступа: private, public, protected.
6. Простое и множественное наследование.
7. Конструкторы и деструкторы класса.
8. Перегрузка функций. Перегрузка конструктора.
9. Перегрузка операторов. Перегрузка бинарных и унарных операторов.
10. Динамический полиморфизм.
11. Абстрактные классы.
12. Шаблоны классов.
13. Шаблоны функций.
14. Библиотека стандартных шаблонов.

**7. Учебно-методическое и информационное обеспечение дисциплины**  
**7.1. Рекомендуемая литература и источники информации (основная и до-  
 полнительная)**

Согласовано  
 Зав. библиотекой ФГБОУ ВО  
 «ДГТУ»

\_\_\_\_\_  
 Подпись

№	Виды занятий (лк, лб, срс)	Комплект необходимой учебной литературы по дисциплинам (наименование учебника, пособия)	Авторы	Издательство и год издания	Кол-во пособий, учебников и прочей литературы	
					в библи	на каф
<b>О С Н О В Н А Я</b>						
1.	Лк, лб, срс	Основы объектно-ориентированного программирования: учебное пособие	Букунов, С. В., О. В. Букунова	СПб. : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017. — 196 с.	Режим доступа: <a href="http://www.iprbookshop.ru/74339.html">http://www.iprbookshop.ru/74339.html</a>	
2.	Лк, лб, срс	Объектно-ориентированное программирование. Часть 1: лабораторный практикум	Николаев, Е. И.	Ставрополь : Северо-Кавказский федеральный университет, 2015. — 183 с.	Режим доступа: <a href="http://www.iprbookshop.ru/62966.html">http://www.iprbookshop.ru/62966.html</a>	
3.	Лк, лб, срс	Объектно-ориентированное программирование. Часть 2: лабораторный практикум	Николаев, Е. И.	Ставрополь : Северо-Кавказский федеральный университет, 2015. — 156 с.	Режим доступа: <a href="http://www.iprbookshop.ru/63218.html">http://www.iprbookshop.ru/63218.html</a>	

4.	Лк, лб, срс	Объектно-ориентированное программирование: учебное пособие	Николаев, Е. И.	Ставрополь : Северо-Кавказский федеральный университет, 2015. — 225 с.	Режим доступа: <a href="http://www.iprbookshop.ru/62967.html">http://www.iprbookshop.ru/62967.html</a>	
5.	Лк, лб, срс	Объектно-ориентированное программирование на С++	Васильев, А. Н.	СПб. : Наука и Техника, 2016. — 544 с.	Режим доступа: <a href="http://www.iprbookshop.ru/60648.html">http://www.iprbookshop.ru/60648.html</a>	
6.	Лк, лб, срс	Технологии объектно-ориентированного программирования: учебное пособие для ВУЗов.	Хорев П.Б.	М.:Изд.дом «Академия», 2008	30	1
<b>ДОПОЛНИТЕЛЬНАЯ</b>						
7.	Лк, лб, срс	Объектно-ориентированное программирование: практикум: учеб. пособие для вузов	Павловская Т.А., Щупак Ю.А.	СПб.: Питер, 2004, 265 с.	-	1
8.	Лк, лб, срс	Как программировать на С++. 5-е издание	Дейтел Х.М., Дейтел П.Ж.	М.: Изд. «БИНОМ», 2008	1	-
9.	Лк, лб, срс	Основы программирования: учебник для студ.	Семакин И.Г., Шестаков А.П.	М.:Изд.центр «Академия», 2007	30	1
10.	Лк, лб, срс	Язык программирования Си++. Курс лекций. Учебное пособие/Издание второе, исправленное	Фридман А.Л.	М.: ИНТУ-ИТ.РУ «Интернет-университет Информационных технологий», 2004, 264с.	1	-

11.	Лк, лб, срс	Язык Си++. Учебное пособие. 5 издание.	Подбельский В.В.	М.:Финансы и статистика, 2001г	15	1
12.	Лк, лб, срс	Эффективное использование STL/C	Мейерс С.	СПб.: Питер, 2002	2	1
13.	Лк, лб, срс	Программирование на Visual C++	Глушаков С.В., Коваль А.В., Смирнов С.В.	2003, 728с	1	-
14.	Лк, лб, срс	Философия Java. Библиотека программиста	Эккель Б.	СПб.: Питер, 2001, 880с.	1	-
15.	Лк, лб, срс	C++. Объектно-ориентированное программирование. Задачи и упражнения.	Лаптев В.В., Морозов А.В., Бокова А.В.	СПб.: Питер, 2007, 288с.	-	1
16.		C/C++. Программирование на языке высокого уровня	Павловская Т.А.	СПб.: Питер, 2007, 461с.	-	1

## 7.2. Программное и информационное обеспечение

Интегрированные среды разработки программ Microsoft Visual Studio, базы данных, информационно – справочные и поисковые системы; вузовские электронно-библиотечные системы учебной литературы; база научно-технической информации ВИНТИ РАН.

Интернет-ресурсы:

- <http://www.iprbookshop.ru/> - Электронно-библиотечная система «IPR BOOKS».
- [www.e.lanbook.com](http://www.e.lanbook.com) - Электронно-библиотечная система «Издательство «Лань».
- <http://www.edu.ru/> – Федеральный портал «Российское образование».
- <http://window.edu.ru/> – Единое окно доступа к образовательным ресурсам.
- <http://www.studentlibrary.ru/> – электронно-библиотечная система «Консультант Студента».
- <http://elibrary.ru/> – научная электронная библиотека.



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

Для проведения лекционных занятий на факультете имеется комплект технических средств обучения в составе:

- интерактивная доска;
- переносной компьютер (в конфигурации не хуже: процессор IntelCore 2 Duo, 2 Гбайта ОЗУ, 500 Гбайт НЖМД);
- проектор (разрешение не менее 1280x1024);

Для проведения практических занятий имеется компьютерный класс, оборудованный компьютерами с установленным программным обеспечением, предусмотренным программой дисциплины.

Программа составлена в соответствии с требованиями ФГОС ВО с учетом рекомендаций и примерной ООП ВО по направлению 01.03.02 «Прикладная математика и информатика» и профилю подготовки «Системное программирование и компьютерные технологии».

Рецензент от выпускающей кафедры (работодателя) по направлению

(специальности) \_\_\_\_\_

подпись,

ФИО