

**Министерство образования и науки Российской Федерации**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего профессионального образования**  
**«Владимирский государственный университет**  
**имени Александра Григорьевича и Николая Григорьевича Столетовых»**  
**(ВлГУ)**



УТВЕРЖДАЮ  
 Проректор  
 по учебно-методической работе

А.А.Панфилов

« 17 » \_\_\_\_\_ 20 15 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ**  
 \_\_\_\_\_  
 (наименование дисциплины)

Направление подготовки: 02.03.03 Математическое обеспечение и администрирование информационных систем

Профиль/программа подготовки:

Уровень высшего образования: бакалавриат

Форма обучения: очно-заочная, ускоренная

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточного контроля (экз./зачет)
2	1 ЗЕ / 36 час.	–	–	–	36	зачёт (переаттестация)
3	3 ЗЕ / 108 час.	18	–	18	36	КР, экзамен (36 ч.)
Итого	4 ЗЕ / 144 час.	18	–	18	72	зачёт (переаттестация), КР, экзамен (36 ч.)

## 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Важным фактором подготовки специалистов в области математического обеспечения и администрирования информационных систем является умение программировать, используя современные языки, включающие объектные возможности, знакомство с основными методами и современными технологиями программирования, в том числе с использованием объектных библиотек конкретных языков.

Цель курса: на базе одного из самых распространенных языков программирования высокого уровня C++, позволяющего использовать объектно-ориентированное программирование, познакомиться с общепринятыми методиками создания приложений в современных объектно-ориентированных программных средах.

Цель обусловила и задачи курса: приобретение теоретических знаний и практических навыков объективно-ориентированного программирования прикладных задач на языке C++ в интегрированной среде программирования MS Visual Studio, в том числе с использованием стандартных библиотек классов. При практической работе на ЭВМ рекомендуются в качестве прикладных задач типовые вычислительные задачи, решение которых на предыдущем курсе выполнялось с использованием других подходов.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Объектно-ориентированное программирование» относится к обязательным дисциплинам вариативной части ОПОП подготовки бакалавров по направлению «Математическое обеспечение и администрирование информационных систем».

Дисциплина изучается в третьем семестре и требует освоения следующих курсов:

- Алгоритмы и анализ сложности
- Основы программирования
- Иностранный язык

В рамках перечисленных дисциплин студенты получают следующие знания и навыки, необходимые для освоения курса «Объектно-ориентированное программирование»:

- Умение применять методики алгоритмизации задач, выбирать наиболее эффективные алгоритмы
- Знание основных современных средств разработки программных продуктов
- Умение получать информацию из источников на иностранном языке

По дисциплине предусмотрена частичная переаттестация в объёме 1 зачётной единицы, так как часть содержания курса студенты осваивают в ходе обучения на профильных специальностях среднего профессионального образования.

Дисциплина формирует знания и навыки, необходимые в ходе дальнейшего обучения и в практической деятельности квалифицированного специалиста. В рамках учебного процесса может быть использована в ходе изучения дисциплин:

- Вычислительная математика
- Структуры и алгоритмы компьютерной обработки данных
- Базы данных
- Программирование для Интернет и веб-дизайн
- Функциональное программирование
- Разработка кросс-платформенных приложений на языке Java
- Рекурсивно-логическое программирование
- Защита информации / Безопасность информационных систем
- Распределённая обработка информации / Параллельное программирование

Кроме того, полученные навыки будут необходимы при прохождении практик и подготовке выпускной квалификационной работы.

### 3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате изучения дисциплины студент должен частично овладеть следующими компетенциями:

- ОПК-3. готовностью анализировать проблемы и направления развития технологий программирования
- ОПК-4. способностью применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения
- ОПК-7. способностью использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений
- ОПК-11. готовностью использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

#### 1) Знать:

- Основные концептуальные положения объектно-ориентированного направления программирования, методы, способы и средства разработки программ в рамках этого направления (ОПК-7)
- Основные проблемы объектно-ориентированного подхода и тенденции развития объектно-ориентированных языков программирования (ОПК-3)

#### 2) Уметь:

- Проектировать систему классов для заданной предметной области (ОПК-7, ОПК-11)
- Программно реализовывать разработанную систему классов для заданной предметной области (ОПК-4, ОПК-11)
- Анализировать эффективность объектно-ориентированного программного кода (ОПК-11)

#### 2) Владеть

- Средствами разработки объектно-ориентированного программного кода (ОПК-4)

### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 часа.

№ п/п	Раздел (тема) дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем учебной работы, с применением интерактивных методов (в часах / %)	Формы текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	Контрольные работы	СРС	КП / КР		
1	Раздел 1. Парадигма	2	1-6	-	-	-	-	10	-	-	-

	объектно-ориентированного программирования										
2	Раздел 2. Абстракция и инкапсуляция	2	1-6	-	-	-	-	16	-	-	
3	Раздел 3. Одиночное наследование и полиморфизм	2	1-6	-	-	-	-	10	-	-	
<b>Итого семестр 2</b>		-	-	-	-	-	-	<b>36</b>	-	-	<b>Переаттестация (зачет)</b>
4	Раздел 1. Парадигма объектно-ориентированного программирования	3	1-2	2	-	-	-	4	-	-	Рейтинг-контроль №1
5	Раздел 2. Абстракция и инкапсуляция	3	1-6	4	-	6	-	8	-	3 / 30%	
6	Раздел 3. Одиночное наследование и полиморфизм	3	7-12	6	-	6	-	12	-	3 / 25%	Рейтинг-контроль №2
7	Раздел 4. Множественное наследование	3	13-18	6	-	6	-	12	-	3 / 25%	Рейтинг-контроль №3
<b>Итого семестр 3</b>				<b>18</b>		<b>18</b>		<b>36</b>	<b>КР</b>	<b>9 / 25%</b>	<b>экзамен (36 ч)</b>
<b>Всего</b>		<b>2</b>	<b>-</b>	<b>18</b>	<b>-</b>	<b>18</b>	<b>-</b>	<b>72</b>	<b>КР</b>	<b>9 / 25%</b>	<b>Переаттестация (зачет), Экзамен (36 ч)</b>

### Темы лекций.

#### **Раздел 1. Парадигма объектно-ориентированного программирования.**

1. Общая качественная характеристика ООП, основные свойства и возможности. Составной тип класс: объявление, структура, возможные составляющие компоненты; объект типа класс. Примеры.

#### **Раздел 2. Абстракция и инкапсуляция.**

1. Метки видимости. Доступ к открытым и закрытым членам.
2. Конструкторы и деструкторы: назначение, варианты объявления и использования (в том числе создание и разрушение объектов в динамической памяти).
3. Статистические члены класса. Друзья класса.

#### **Раздел 3. Одиночное наследование и полиморфизм.**

1. Иерархия классов: базовый класс, производный класс. Модификаторы доступа. Доступ к членам базовых и производных классов "изнутри" производственного класса и "извне".
2. Указатели на объекты классов. Преобразование типов для объектов и указателей по иерархии наследования, возможные операции присваивания.
3. Программные возможности распознавания типа, к которому принадлежит указуемый объект (при работе по указателям). Механизм виртуальных функций как вариант позднего связывания объекта с методами: особенности компиляции типа (таблицы виртуальных методов), объектов, вызов методов в этой ситуации (через объекты, через указатели разных типов).

4. Перегрузка операторов для классов (++ , — в префиксном и постфиксном исполнении; +, -, =; « и »); последние две операции на стандартных потоках ввода/вывода).

#### **Раздел 4. Множественное наследование.**

1. Множественное наследование и виртуальные базовые классы.

##### **Лабораторный практикум**

- 1) Абстракция и инкапсуляция (2 ч).
- 2) Перегрузка операций (4 ч).
- 3) Одиночное наследование и полиморфизм (6 ч).
- 4) Множественное наследование (6 ч).

### **5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ**

В рамках лекционного курса:

- Технология проблемного обучения (case study). При рассмотрении вопросов практического применения рассмотренного теоретического материала, используется диалог со студентами на предмет возможных способов решения поставленной задачи. Особенно активно данная технология применяется в рамках четвертого раздела курса.

В рамках лабораторного практикума:

- Технология уровневой дифференциации. Прежде всего, при постановке заданий на лабораторные работы, а также при проведении контрольных мероприятий и при проведении экзамена.

В рамках курсового проектирования и самостоятельной работы:

- Метод проектов. В группах по 2-3 человека студенты на практике осваивают принципы объектно-ориентированного программирования.

### **6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ**

#### **1 семестр**

##### **Самостоятельная работа студентов (при подготовке к переаттестации)**

Состоит в восстановлении знаний и умений студента в области объектно-ориентированного программирования, полученных при обучении на предыдущем уровне образования. При этом должны быть рассмотрены вопросы:

1. Составные типы данных. Структуры.
2. Классы и объекты.
3. Методы и сообщения.
4. Поля и свойства.
5. Указатель this.
6. Наследование свойств классов.
7. Размещение объектов в памяти.
8. Работа с указателями
9. Ввод/вывод с логических устройств в C++ через объекты стандартных классов; потоки с дисковых файлов.

##### **Рейтинг-контроли:**

##### **Рейтинг-контроль №1**

Призван вывить знания студентами принципов инкапсуляции и абстракции в объектно-ориентированном программировании.

Список заданий:

- 1) *Составной тип класс: объявление, структура, возможные составляющие компоненты; объект типа класс. Примеры.*
- 2) *Метки видимости. Доступ к открытым и закрытым членам. Указатель this. Примеры.*

- 3) *Конструкторы и деструкторы: назначение, варианты объявления и использования (в том числе создание и разрушение объектов в динамической памяти).*
- 4) *Статистические члены класса. Примеры. Друзья класса: назначение и примеры использования.*
- 5) *Перегрузка операторов для классов (++ , — в префиксном и постфиксном исполнении; +, -, =; « и »; последние две операции на стандартных потоках ввода/вывода).*
- 6) *Ввод/вывод с логических устройств в C++ через объекты стандартных классов; потоки с дисковых файлов.*
- 7) *Шаблоны классов. "Вмещающие" классы. Примеры.*

### **Рейтинг-контроль №2**

Призван вывить знания студентами принципов одиночного наследования и полиморфизма в объектно-ориентированном программировании.

Список заданий:

- 1) *Иерархия классов: базовый класс, производный класс.*
- 2) *Модификаторы доступа при наследовании. Доступ к членам базовых и производных классов "изнутри" производственного класса и "извне".*
- 3) *Указатели на объекты классов. Преобразование типов для объектов и указателей по иерархии наследования, возможные операции присваивания.*
- 4) *Общепринятые соглашения об именах произвольных типов (ссылок и указателей). Доступ к одноименным членам классов по иерархии наследования, механизм раннего связывания.*
- 5) *Механизм виртуальных функций как вариант позднего связывания объекта с методами: особенности компиляции типа (таблицы виртуальных методов), объектов, вызов методов в этой ситуации (через объекты, через указатели разных типов).*
- 6) *Виртуальные деструкторы. Абстрактные классы.*
- 7) *Указатели на члены класса (поля и методы): объявление типов, переменных, примеры и рекомендации для использования.*

### **Рейтинг-контроль №3**

Призван вывить знания студентами принципов множественного наследования.

Список заданий:

- 1) *Множественное наследование. Правила доступа к полям и методам.*
- 2) *Множественное наследование и виртуальные базовые классы.*
- 3) *Полиморфизм при множественном наследовании.*
- 4) *Работа с указателями на базовый класс при множественном наследовании.*
- 5) *Проблемы множественного наследования.*
- 6) *Примеры применения множественного наследования.*

### **Зачёт (переаттестация)**

Призван вывить знания студентами базовых принципов объектно-ориентированного подхода к программированию. Проводится в форме решения задачи на любом из известных студенту языков, позволяющих использовать объектно-ориентированный подход. Примеры задач:

1. *Описать класс TDevice, инкапсулирующий следующую информацию об электронном устройстве: длина, высота, ширина, масса, название, потребляемая мощность. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для любых двух свойств описать по паре методов, предназначенных для изменения значения свойства и для получения этого значения.*
2. *Описать класс TAuto, инкапсулирующий следующую информацию об автомобиле: длина, высота, ширина, масса, марка, мощность двигателя. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для любых двух свойств описать по паре методов, предназначенных для изменения значения свойства и для получения этого значения.*
3. *Описать класс TRoom, инкапсулирующий следующую информацию о комнате: длина, высота, ширина, цвет обоев, материал напольного покрытия, количество дверей. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для любых двух свойств описать по паре методов, предназначенных для изменения значения свойства и для получения этого значения.*

4. *Описать класс TComputer, инкапсулирующий следующую информацию о компьютере: архитектура, процессор, объём ОЗУ, объём винчестера, мощность блока питания, производитель. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для любых двух свойств описать по паре методов, предназначенных для изменения значения свойства и для получения этого значения.*

#### **Курсовая работа и самостоятельная работа студентов при освоении основного раздела курса**

Следует отметить, что в связи с существенной практической направленностью курса, наибольший вес имеют результаты, полученные студентами в рамках курсового проектирования.

В ходе выполнения курсовой работы необходимо разработать оконное приложение, тестирующее систему классов, спроектированную и реализованную студентами для решения конкретной задачи из некоторой предметной области.

Примерная тематика курсовых работ:

- 1) Библиотека классов, реализующих графические примитивы на плоскости с возможностью аффинных преобразований.
- 2) Библиотека классов, реализующих графические примитивы на плоскости с реализацией операций над множествами.
- 3) Библиотека классов, реализующих графические примитивы в трёхмерном пространстве с возможностью аффинных преобразований.
- 4) Классы для описания оконного графического интерфейса, аналогичного интерфейсу MS Windows. Должны быть реализованы обработчики событий с использованием виртуальных функций.
- 5) Библиотека классов, реализующих комплексную арифметику.
- 6) Библиотека классов, реализующих векторы в  $n$ -мерном пространстве.
- 7) Реализация строк и операций над ними, включая работу с регулярными выражениями.
- 8) «Обобщённый массив» (позволяющий хранить данные произвольных типов).
- 9) Ассоциативный массив (хэши-массив) (с возможностью хранения данных произвольных типов).
- 10) Реализация различных типов графов и операций над ними.
- 11) Система классов для обеспечения работы с абонентами телефонной компании.
- 12) Система классов для обеспечения работы деканата.
- 13) Система классов, описывающих сотрудников предприятия/организации с их функциями (сотрудник, менеджер, ...).
- 14) Система классов, описывающих различные транспортные средства.
- 15) Моделирование замкнутой биологической системы (корм, травоядное, хищник).
- 16) Моделирование муравейника (несколько типов муравьёв, источники питания, внешние раздражители, ...).
- 17) Моделирование дорожного движения на заданной карте дорог.
- 18) Моделирование компьютерной сети (стационарной).
- 19) Моделирование компьютерной сети с изменяющейся топологией.
- 20) Моделирование планетарной системы.
- 21) Система классов для реализации матричной арифметики.
- 22) Библиотека классов для реализации игрового приложения.

Для выполнения курсовой работы студентам необходимо самостоятельно изучить следующие вопросы:

1. Шаблоны классов. "Вмещающие" классы. Примеры.
2. Указатели на объекты классов. Преобразование типов для объектов и указателей по иерархии наследования, возможные операции присваивания.
3. Общепринятые соглашения об именах произвольных типов (ссылок и указателей). Доступ к одноименным членам классов по иерархии наследования, механизм раннего связывания.
4. Программные возможности распознавания типа, к которому принадлежит указуемый объект (при работе по указателям).
5. Таблицы виртуальных методов.
6. Виртуальные деструкторы. Абстрактные классы.

7. Базовые конструкции языка C#
8. Объектная модель языка C#
9. Особенности и перспективы использования и развития языка C#

#### Экзамен (контроль освоения всех разделов курса)

Предполагает устный ответ на один теоретический вопрос и письменное решение студентом практической задачи: написание программ на языках C++ или C#. Ответ на устный вопрос должен быть обязательно проиллюстрирован примерами программного кода. Конкретная задача выбирается случайным образом из базы задач, имеющейся у преподавателя. Решение задачи должно быть написано на обоих языках. Во время решения задачи студент может пользоваться справочной литературой и конспектами. Уровень решения определяет выставляемую оценку. При этом студентом предварительно должны быть выполнены все задания лабораторного практикума.

Теоретические вопросы.

- 1) Составной тип класс: объявление, структура, возможные составляющие компоненты; объект типа класс. Примеры.
- 2) Метки видимости. Доступ к открытым и закрытым членам. Указатель *this*. Примеры.
- 3) Конструкторы и деструкторы: назначение, варианты объявления и использования.
- 4) Конструктор инициализации.
- 5) Дружественные функции и дружественные классы: назначение и примеры использования.
- 6) Перегрузка операций для классов.
- 7) Приведение типов (явное и неявное).
- 8) Одиночное наследование: базовый класс, производный класс, правила доступа к полям и методам.
- 9) Механизм виртуальных функций как реализация полиморфизма. Абстрактные классы.
- 10) Шаблонные классы.
- 11) Множественное наследование. Правила доступа к полям и методам. Порядок вызова конструкторов.
- 12) Множественное наследование. Виртуальные базовые классы. Полиморфизм при множественном наследовании.
- 13) Особенности языка C# в сравнении с языком C++.

Примеры задач:

1. Описать класс *TDisplay*, инкапсулирующий следующую информацию о компьютерном мониторе: производитель, модель, размер диагонали, масса, потребляемая мощность. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для одного из свойств описать пару методов, предназначенных для изменения значения свойства и для получения этого значения. Описать класс *TTouchpad*, описывающий сенсорный дисплей и являющийся наследником класса *TDisplay*. Набор дополнительных свойств этого класса определить самостоятельно. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор.
2. Описать класс *TPlane*, инкапсулирующий следующую информацию о самолёте: длина, размах крыльев, масса, максимальная дальность перелёта. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для одного из свойств описать пару методов, предназначенных для изменения значения свойства и для получения этого значения. Описать класс *TBoat*, описывающий самолёт-бомбардировщик и являющийся наследником класса *TPlane*. Набор дополнительных свойств этого класса определить самостоятельно. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор.
3. Описать класс *TDog*, инкапсулирующий следующую информацию о собаке: порода, кличка, высота в холке, вес, длина хвоста. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для одного из свойств описать пару методов, предназначенных для изменения значения свойства и для получения этого значения. Описать класс *TSheepdog*, описывающий пастушью собаку и являющийся



наследником класса *TDog*. Набор дополнительных свойств этого класса определить самостоятельно. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор.

4. Описать класс *TCat*, инкапсулирующий следующую информацию о кошке: кличка, длительность сна в день, возраст, вес. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор. Для одного из свойств описать пару методов, предназначенных для изменения значения свойства и для получения этого значения. Описать класс *TTiger*, описывающий дрессированного тигра и являющийся наследником класса *TCat*. Набор дополнительных свойств этого класса определить самостоятельно. У класса должен быть, как минимум, один конструктор с параметрами. При необходимости должен быть описан деструктор.

## 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература:

1. Языки программирования. Концепции и принципы [Электронный ресурс] / Кауфман В.Ш. - М.: ДМК Пресс, 2010. ISBN 978-5-94074-622-5
2. Программирование на языке C++: Учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 512 с.: ил.; 60x90 1/16 + CD-ROM. - (Проф. обр.). (п, cd rom) ISBN 978-5-8199-0492-3
3. 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

б) дополнительная литература:

1. Практикум по объектно-ориентированному программированию [Электронный ресурс] / И. А. Бабушкина, С.М. Окулов.-4-е изд. (эл).-Электрон. текстовые дан. (1 файл pdf : 369 с.). - М. : БИНОМ. Лаборатория знаний, 2015. - ISBN 978-5-9963-2542-9
2. Эффективное использование C++. 55 верных способов улучшить структуру и код ваших программ - М.: ДМК Пресс, 2008. - 300 с.: ил. - ISBN 5-94074-304-8
3. Программирование на языке высокого уровня. Программирование на языке Object Pascal: Учебное пособие / Т.И. Немцова; Под ред. Л.Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2015. - 496 с.: ил.; 60x90 1/16. - (ПО). (cd rom) ISBN 978-5-8199-0372-8

в) периодические издания:

1. Computerworld Россия, ISSN: 1560-5213.
2. Мир ПК, ISSN: 0235-3520.

в) интернет-ресурсы:

1. MSDN. Возвращение к C++ (современный C++). // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/hh279654.aspx>
2. MSDN. Справочник по языку C++. // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/3bstk3k5.aspx>
3. MSDN. Руководство по программированию на C#. // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>
4. MSDN. Справочник по C#. // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/618ayhy6.aspx>
5. MSDN. Visual Studio. // Режим доступа: <https://www.visualstudio.com/>

## 8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Компьютерные классы для проведения лабораторных занятий. Компьютеры должны иметь выход в Интернет для получения доступа к справочной информации Microsoft Developer Network.

Среда разработки Microsoft Visual Studio.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 02.03.03 Математическое обеспечение и администрирование информационных систем

Рабочую программу составил доцент каф. ФиПМ Лексин А.Ю.  
(ФИО, подпись)

Рецензент

(представитель работодателя)

ген. директор ООО "РС Сервис" Д.Квасов Д.С.  
(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФиПМ

Протокол № 11А от 17.04.15 года

Заведующий кафедрой

(ФИО, подпись)

С.М. Франкман

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 02.03.03

Протокол № 11А от 17.04.15 года

Председатель комиссии

(ФИО, подпись)

С.М. Франкман

### ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Рабочая программа одобрена на \_\_\_\_\_ учебный год

Протокол заседания кафедры № \_\_\_\_\_ от \_\_\_\_\_ года

Заведующий кафедрой \_\_\_\_\_

Рабочая программа одобрена на \_\_\_\_\_ учебный год

Протокол заседания кафедры № \_\_\_\_\_ от \_\_\_\_\_ года

Заведующий кафедрой \_\_\_\_\_

Рабочая программа одобрена на \_\_\_\_\_ учебный год

Протокол заседания кафедры № \_\_\_\_\_ от \_\_\_\_\_ года

Заведующий кафедрой \_\_\_\_\_