МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
(МИНОБРНАУКИ РОССИИ)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ДонГТУ»)

Факультет информационных технологий и автоматизации производственных процессов

Кафедра интеллектуальных систем и информационной безопасности

УТВЕРЖДАЮ
И.о. проректора по учебной работе
_____Д.В. Мулов

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Объектно-ориентированное программирование
(наименование дисциплины)

10.05.03 Информационная безопасность автоматизированных систем
(код, наименование специальности)

Безопасность открытых информационных систем
(специализация)

Квалификация специалист по защите информации
(бакалавр/специалист/магистр)

Форма обучения очная
(очная, очно-заочная, заочная)

Алчевск, 2024

1 Цели и задачи изучения дисциплины

Цели дисциплины. Целью изучения дисциплины «Основы программирования» является выработка у студентов передовых научно-технических воззрений, ориентации их на мировой уровень производительности труда, подготовке специалистов, которые должны обеспечить разработку качественных компьютерных программ, снижение материальных затрат, сокращение сроков проектирования.

Задачи изучения дисциплины:

- приобретение студентами знаний, умений и навыков в области алгоритмизации, программирования, основным этапам решения задач на ЭВМ;
- ознакомление с основами программирования на языке С++;
- приобретение навыков использовать полученные знания в прикладных исследованиях, проектировании и эксплуатации информационных систем и технологий;
- умение создавать программные приложения для проведения инженерных расчетов, обработки числовой, текстовой и графической информации.

Дисциплина направлена на формирование общепрофессиональной (ОПК-7) компетенции выпускника.

2 Место дисциплины в структуре образовательной программы

Логико-структурный анализ дисциплины – курс входит в БЛОК 1 «Дисциплины (модули)», часть, формируемую участниками образовательных отношений подготовки студентов по специальности 10.05.03 «Информационная безопасность автоматизированных систем» (10.05.03-05 «Безопасность открытых информационных систем»).

Дисциплина реализуется кафедрой специализированных компьютерных систем. Основывается на базе дисциплин: «Информатика», «Основы алгоритмизации», «Основы программирования».

Является основой для изучения следующих дисциплин: «Вычислительная математика», «Базы данных», «Безопасность систем баз данных».

Для изучения дисциплины необходимы компетенции, сформированные у студента для решения профессиональных задач деятельности, связанных с разработкой программного обеспечения.

Курс является фундаментом для ориентации студентов в сфере разработки программного обеспечения информационных систем.

Общая трудоемкость освоения дисциплины составляет 5 зачетных единиц, 180 ак.ч. Программой дисциплины предусмотрены лекционные (36 ак.ч.), лабораторные (36 ак.ч.), практические (18 ак.ч.) занятия и самостоятельная работа студента (90 ак.ч.).

Дисциплина изучается на 2 курсе в 4 семестре. Форма промежуточной аттестации – экзамен. По дисциплине предусмотрена курсовая работа.

3 Перечень результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины «Объектно-ориентированное программирование» направлен на формирование компетенции, представленной в таблице 1.

Таблица 1 –Компетенции, обязательные к освоению

Содержание компетенции	Код компетенции	Код и наименование индикатора достижения компетенции
Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	ОПК-7	ОПК-7.2. Применяет методы и инструментальные средства программирования для решения профессиональных задач

4 Объём и виды занятий по дисциплине

Общая трудоёмкость учебной дисциплины составляет 5 зачётных единицы, 180 ак.ч.

Самостоятельная работа студента (СРС) включает проработку материалов лекций, подготовку к практическим занятиям, текущему контролю, выполнение индивидуального задания, самостоятельное изучение материала и подготовку к экзамену, выполнение курсовой работы.

При организации внеаудиторной самостоятельной работы по данной дисциплине используются формы и распределение бюджета времени на СРС для очной формы обучения в соответствии с таблицей 2.

Таблица 2 – Распределение бюджета времени на СРС

Вид учебной работы	Всего ак.ч.	Ак.ч. по семестрам
		4
Аудиторная работа, в том числе:	90	90
Лекции (Л)	36	36
Практические занятия (ПЗ)	-	-
Лабораторные работы (ЛР)	36	36
Курсовая работа/курсовой проект	18	18
Самостоятельная работа студентов (СРС), в том числе:	90	90
Подготовка к лекциям	9	9
Подготовка к лабораторным работам	18	18
Подготовка к практическим занятиям / семинарам	9	9
Выполнение курсовой работы / проекта	20	20
Расчетно-графическая работа (РГР)	-	-
Реферат (индивидуальное задание)	-	-
Домашнее задание	-	-
Подготовка к контрольным работам	-	-
Подготовка к коллоквиуму	-	-
Аналитический информационный поиск	-	-
Работа в библиотеке	9	9
Подготовка к экзамену (диф.зачету)	25	25
Промежуточная аттестация – экзамен (Э)	Э	Э
Общая трудоемкость дисциплины		
ак.ч.	180	180
з.е.	5	5

5 Содержание дисциплины

С целью освоения компетенции, приведенной в п.3 дисциплина разбита на 2 тем:

– тема 1 (Объектно-ориентированное программирование и проектирование);

– тема 2 (Проектирование визуальных приложений C++).

Виды занятий по дисциплине и распределение аудиторных часов для очной формы приведены в таблице 3.

Таблица 3 – Виды занятий по дисциплине и распределение аудиторных часов (очная форма обучения)

№ п/п	Наименование темы (раздела) дисциплины	Содержание лекционных занятий	Трудоемкость в ак.ч.	Темы практических занятий	Трудоемкость в ак.ч.	Тема лабораторных занятий	Трудоемкость в ак.ч.
1	2	3	4	5	6	7	8
1	Объектно-ориентированное программирование и проектирование	Программные продукты как сложные системы. Признаки сложности. Назначение объектно-ориентированного программирования. Синтаксис и особенности C++. Ошибки и исключительные ситуации. Определение объекта и класса. Объектная декомпозиция, диаграмма классов. Модули. Интерфейсы и реализации. Наследование, полиморфизм и инкапсуляция. Классы в C++. Шаблоны классов, механизм в C++ для его реализации. Изучение библиотек стандартных шаблонов (STL). Шаблоны проектирования. Порождающие шаблоны проектирования, структурные шаблоны проектирования, шаблоны поведения. Принципы SOLID. Тенденции и пути развития ООП	26	-	-	Основы создания консольных и визуальных приложений в QT Модульное программирование Обработка исключительных ситуаций в C++ Создание классов и наследование Создание шаблонов классов в C++.Использование библиотеки шаблонов STL	4 4 4 4 6

Завершение таблицы 3

1	2	3	4	5	6	7	8
2	Проектирование визуальных приложений C++	Библиотека Qt. Иерархия классов Qt. Создание проектов Qt. Основной элемент приложения – widget. Виды элементов управления. Компоновки. Обработка событий. Элементы ввода и отображения информацииУправляющие элементы: кнопки, переключатели, флажки. Разработка интерфейса пользователя: основные принципы и концепции. Элементы управления Qt для разработки пользовательского интерфейса. Однооконный и многооконный интерфейсы. Вывод результатов анализа в виде графиков. Основы графики Qt C++. Основы печати в программах C++.	10	–	–	Визуальные приложения Qt Графика Qt	6 8
Всего аудиторных часов			36	-		36	

Таблица 4 – Виды занятий по дисциплине и распределение аудиторных часов по курсовой работе (очная форма обучения)

№ п/п	Тема занятия	Трудоемкость в ак.ч.
1	Описание задачи, разработка содержательной блок-схемы алгоритма программы	2
2	Определение и описание переменных, констант и подпрограмм (функций)	2
3	Разработка интерфейса программы	2
4	Разработка программного кода основного алгоритма	4
5	Разработка программного кода обработчиков событий	4
6	Выполнение тестового примера	2
7	Оформление пояснительной записки	2
Всего аудиторных часов		18

6 Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации студентов по дисциплине

6.1 Критерии оценивания

В соответствии с Положением о кредитно-модульной системе организации образовательного процесса ФГБОУ ВО «ДонГТУ» (https://www.dstu.education/images/structure/license_certificate/polog_kred_modul.pdf) при оценивании сформированности компетенций по дисциплине используется 100-балльная шкала.Перечень компетенций по дисциплине и способы оценивания знаний приведены в таблице 5.

Таблица 5 – Перечень компетенций по дисциплине и способы оценивания знаний

Код и наименование компетенции	Способ оценивания	Оценочное средство
ОПК-7	Экзамен Диф.зачет	Комплект контролирующих материалов для экзамена

Всего по текущей работе в семестре студент может набрать 100 баллов, в том числе:

– лабораторные работы – всего 100 баллов;

Экзамен проставляется автоматически, если студент набрал в течении семестра не менее 60 баллов и отчитался за каждую контрольную точку. Минимальное количество баллов по каждому из видов текущей работы составляет 60% от максимального.

Экзамен по дисциплине «Объектно-ориентированное программирование» проводится по результатам работы в семестре. В случае, если полученная в семестре сумма баллов не устраивает студента, во время сессии студент имеет право повысить итоговую оценку либо в форме устного собеседования по приведенным ниже вопросам (п.п. 6.5), либо в результате тестирования.

Шкала оценивания знаний при проведении промежуточной аттестации приведена в таблице 6.

Таблица 6 –Шкала оценивания знаний

Сумма баллов за все виды учебной деятельности	Оценка по национальной шкале зачёт/экзамен
0-59	Не зачтено/неудовлетворительно
60-73	Зачтено/удовлетворительно
74-89	Зачтено/хорошо
90-100	Зачтено/отлично

6.2 Домашнее задание

Домашнее задание не предусмотрено.

6.3Темы для рефератов (презентаций) – индивидуальное задание

Реферат (индивидуальное задание) не предусмотрен.

6.4 Оценочные средства для самостоятельной работы и текущего контроля успеваемости

Тема 1. Объектно-ориентированное программирование и проектирование.

1) С помощью чего реализуется принцип полиморфизма в С ++?

Варианты ответа:

а) наличия множественного наследования.

б) наличия виртуальных методов.

в) использование виртуального наследования.

г) наличия абстрактных классов.

2) В программе описан класс и объект

class A {public: int a, b, c; }; A * obj; Как обратиться к атрибуту c?

Варианты ответа:

а) obj.c

б) obj->c

в) obj A -> -> c

г) obj-> A.c

3) Отметьте правильное утверждение для абстрактного класса.

Варианты ответа:

а) класс, у которого все методы чисто виртуальные, называется абстрактным.

б) абстрактный базовый класс навязывает определенный интерфейс всем производным из него классам.

в) невозможно создать объект абстрактного класса.

г) в абстрактном классе не описываются методы вообще.

4) Если в программе на языке С ++ в производном классе переопределена операция new то …

Варианты ответа:

а) все объекты этого класса и все объекты классов, выведенных из него, будут использовать эту операцию независимо от зоне видимости, в которой она переопределена.

б) производные от этого класса могут использовать глобальную операцию применив операцию базовий_класс :: new.

в) операцию new нельзя переопределить.

г) в любом случае эта операция будет доступна только в пределах класса-потомка.

5) Какая функция, не будучи компонентом класса, имеет доступ к его защищенным и внутренних компонентов?

Варианты ответа:

а) шаблонная.

б) полиморфная.

в) дружественная.

г) статическая.

6) Принцип объектно-ориентированного программирования, заключается в объединении атрибутов и методов объекта с целью обеспечения сохранности данных, называется:

Варианты ответа:

а) наследование.

б) сочетание.

в) инициализация.

7) Выберите верное утверждение о деструктор класса в С ++.

Варианты ответа:

а) деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить.

б) деструктор принимает в качестве параметра указатель this.

в) деструктор не содержит параметров.

г) деструктор принимает в качестве параметра тот объект, который нужно уничтожить.

8) Какой из вариантов записи абстрактного класса в С ++ является правильным?

Варианты ответа:

а) abstract class A {virtual int f () = 0;};

б) class A {virtual int f () = 0;};

в) class A {virtual int f () = 0;} abstract;

г) class A {virtual int f ();};

9) Шаблон - это

а) определение нескольких функций (две или больше) с одинаковым именем, но различными параметрами

б) шаблон функции, с помощью которых можно определить алгоритм, который будет применяться к данным различных типов

в) один из способов реализации полиморфизма, заключающийся в возможности одновременного существования в одной области видимости нескольких различных вариантов применения оператора, имеющих одно и то же имя, но различающихся типами параметров, к которым они применяются

9) Что из перечисленного не является модификатором доступа?

а) private

б) public

в) closed

г) protected

Тема 2. Проектирование визуальных приложений С++.

1) Выберите из приведенных примеров виджеты:

а) Кнопка

б) Метка

в) Поле ввода

г) Текстовое поле

д) Числовое поле

2) Какие элементы относятся к элементам ввода информации

а) QListView

б) QDial

в) QLabel

г) QSpinBox

д) QLineEdit

3) С помощью какого класса происходит отрисовка виджетов?

а) QPaint

б) QPainter

в) QDraw

4) Какой класс является базовым для почти всех классов Qt?

а) QObject

б) QWidget

в) QLineEdi

г) QLabel

д) Q_OBJECT

5) Класс событий QKeyEvent отвечает за:

а) Событие для нажатия клавиш клавиатуры

б) Событие для движений мышкой и нажатия клавиш мышки. Посылается виджетам. Выполняется только при нажатии клавиши мышки

в) Посылается окну, которое пользователь пытается закрыть

6) Какой класс отвечает за диалоговое окно?

а) QDialog

б) QMainWindow

в) QWidget

7) Объекты, которые наследуют QObject :

а) имеют имя, которое используется в Qt для реализации различных возможностей

б) могут занимать место в иерархии других объектов QObject

в) могут иметь сигнальнослотовые соединения с другими объектами QObject

г) все перечисленное.

8) Какие в Qt существуют стандартные диалоги?

а) QInputDialog

б) QQuestionDialog

в) QFileDialog

г) QMessageBox

9) Какой метод делает визуальный элемент невидимым?

а) setVisible(false)

б) unVisible(true)

в) setEnabled(true)

6.5 Вопросы для подготовки к экзамену

1. Как охарактеризовать программные продукты как сложные системы? Признаки сложных систем. Декомпозиция. В чем причины сложности программных систем?
2. Что такое объект? Что не является объектом, каковы подходы к выделению объектов? Какие Вы знаете виды отношений между объектами.
3. Что такое модуль в программе?.Что такое интерфейс, класс, тип?
4. Каковы этапы разработки программных средств с использованием объектно-ориентированного подхода?
5. Каковы принципы объектно-ориентированного представления программных систем?
6. Что такое полиморфизм и инкапсуляция?
7. Как выглядит диаграмма классов? Каковы виды отношений между классами?
8. Как осуществляется обработка исключительных ситуаций в C++?
9. Каковы дополнительные принципы ООП?
10. Что такое многопоточность? Как производится синхронизация потоков?
11. Что такое мьютексы и как они реализованы в библиотеке STL?
12. Что такое устойчивость, области видимости и типы переменных?
13. Что такое многопоточное программирование в C++?
14. Какие Вы знаете виды методов в C++? Что такое раннее и позднее связывание?
15. Что такое виртуальные методы? Каково их практическое применение? Что такое таблица виртуальных методов?
16. Что такое конструкторы и как с их помощью производится инициализация объектов? Что такое конструкторы и методы создания экземпляра класса?
17. Что включает библиотека стандартных классов (STL)?
18. Как производится преобразование типов в C++ с помощью dynamic_cast, static_cast, reinterpret_cast, const_cast?
19. Что такое шаблоны классов, template? Какие виды шаблонов Вы знаете?
20. Какие Вы знаете контейнеры STL и их основные методы?
21. Какие классы входят в стандартную библиотеку STL?
22. В чем заключаются принципы SOLID?
23. В чем заключается тестирование?
24. Что такое рефакторинг? Как он осуществляется?
25. Из чего состоит объектная структура предметной области? Что такое объект-значение и агрегат?
26. Как выглядит диаграмма состояний и диаграмма взаимодействия (диаграмма последовательностей)?
27. Что такое метрики оценки программного кода?

28. Что такое форма? Каковы архитектура формы? Каковы её основные свойства?

29. Как создать форму во время исполнения программы? Как сделать так, чтобы окно программы не отображалось на панели задач? Как сделать так, чтобы форма «перехватывала» все нажатия клавиш? Что такое модальная форма? Как определить, каким образом была закрыта модальная форма?

30. Какие свойства отвечают за цвет фона и параметры шрифта компонента управления?

31. Какие Вы знаете события мыши? Как их обработать? Какова последовательность возникновения событий при нажатии и последующем отпускании клавиши мыши?

32. Для чего нужен компонент QLabel? Каковы его основные свойства?

33. Какие Вы знаете события клавиатуры? Как их обработать? Последовательность возникновения событий при нажатии и последующем отпускании клавиш?

34. Для чего нужен компонент QListBox? Каковы его основные свойства? Как задать их на этапе проектирования и выполнения программы? Как определить, какой элемент списка было выбран? Как добавить строку к списку и удалить строку из списка? Привести примеры.Как с помощью программного кода добавлять и удалять строки в списке QListBox?

35. Для чего нужен компонент QComboBox. Каковы его основные свойства, определяющие внешний вид? Как задать значения свойств на этапе проектирования и выполнения программы? Как определить, какой элемент списка было выбран? Как добавить строку к списку и удалить строку из списка? Привести примеры.

36. Что задают свойства Size и Position? Привести пример чтения и программно установки указанных свойств.

37. Для чего нужен диалог QFileDialog? Каковы его основные свойства? Пример использования.

38. Как с помощью программного кода изменить фоновый цвет элемента управления и цвет шрифта, отображающего надпись на нем?

39. Для чего нужен диалог QFontDialog? Каковы его основные свойства? Пример использования.

40. Для чего нужен диалог QColorDialog. Каковы его основные свойства? Пример использования.

41. Для чего нужна строка состояния QStatusBar. Каковы её основные свойства? Как работать с разделами строки состояния?

42. Какое свойство задает текст надписи на элементе управления? Как с помощью программного кода изменить надпись на элементе управления? Какие свойства отвечают за цвет фона и параметры шрифта компонента управления? Как программно задать параметры шрифта?

43. Для чего нужна панель инструментов QToolBar? Каковы её основные свойства? Как работать с кнопками?

44. Для чего нужен элемент управления QCalendarWidget? Каковы его основные свойства? Привести пример использования.

45. Как реализовать контекстные меню в Qt? Каковы его основные свойства?

46. Для чего нужен класс QDateTime? Каковы его основные свойства? Как задать начальную дату?

47. Как создать главное меню в Qt?

48. Для чего нужен компонент QRadioButton? Каковы его основные свойства?

49. Для чего нужен элемент управления QPinBox? Каковы его основные свойства? В чем особенности настройки?

50. Для чего нужна кнопка QButton? Каковы её основные свойства? Как программировать события кнопки? Какие свойства кнопки нужно изменить, чтобы картинка на ней отображалась справа, а текст - слева? Как сделать кнопку «плоской»?

51. Какие элементы управления можно размещать на панели инструментов?

52. Для чего нужен компонент QTextBox? Каковы его основные свойства? Как настроить текстовое поле на ввод многострочного текста? Какие свойства влияют на это? Как обеспечить «прокручивание» содержимого в многострочном текстовом поле? Как задать перенос по словам?

53. Какие элементы управления можно размещать в строке состояния? Привести пример.

54. Для чего нужен компонент QGroupBox? Каковы его основные свойства? Чем они друг от друга отличаются?

55. Какие функции выполняют свойства Visible и Enabled?

56. Что означает свойство Owner? Как его определить? Что означает свойство Parent? Как его определить? На какие другие свойства оно влияет? Как его использовать?

57. Для чего нужен элемент управления «флажок»? Каковы его основные свойства? Как задать их на этапах проектирования и выполнения программы?

58. Графические координаты и доступ к графическому контексту: каковы его основные свойства? Что относится к свойствам объекта Graphics?

59. Для чего нужно средство рисования QPen? Каковы его основные свойства? Пример применения.

60. Для чего нужно средство заполнения QBrush? Каковы его основные свойства? Пример применения.

61. Как рисовать линии и фигуры в Qt.

62. Как произвести масштабирование графиков в программах Qt? Как вывести нескольких графиков в одном графическом контексте?

63. Как произвести вывод текста и графиков на печать в программах Qt? Привести пример.

64. Для чего нужны классы QPrintDialog, QPrinter? Как произвести настройку печати в Qt?

6.6 Примерная тематика курсовых работ

Целью выполнения курсовой работы является углубление и закрепление

студентами знаний основных приемов, методов и принципов работы при решении на ЭВМ задач с использованием языков высокого уровня. Для выполнения работы достаточно знаний основных разделов дисциплины "Объектно-ориентированное программирование". Курсовая работа заключается в разработке модулей подпрограмм и приложений для решения задач заданного класса.

Курсовая работа оформляется в виде пояснительной записки (25-40 стр.), в которой отражаются все этапы создания программного продукта. Обязательными элементами работы являются: объектная модель, диаграмма классов, реализация на языке C++, результаты тестирования.

Индивидуальные варианты заданий

1. Иерархия типов: машиностроительные механизмы.
2. Иерархия типов: строительные машины и механизмы.
3. Иерархия типов: строительный инструмент.
4. Иерархия типов: музыкальные инструменты.
5. Иерархия типов: минералы.
6. Иерархия типов: бабочки.
7. Иерархия типов: насекомые.
8. Иерархия типов: млекопитающие.
9. Иерархия типов: измерительные приборы.
10. Иерархия типов: лекарственные растения.
11. Иерархия типов: библиотека.
12. Иерархия типов: складское хозяйство.
13. Иерархия типов: почтовая служба.
14. Иерархия типов: такси.
15. Иерархия типов: гостиница.
16. Иерархия типов: заказ товаров.
17. Иерархия типов: туристическое агентство.
18. Иерархия типов: портфолио студента.
19. Иерархия типов: успеваемость студентов.
20. Иерархия типов: банкомат.

7. Учебно-методическое и информационное обеспечение дисциплины

7.1 Рекомендуемая литература

Основная литература

1. Фридман, А. Л. Язык программирования C++ : учебное пособие / А. Л. Фридман. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 217 с. [Электронный ресурс]. – URL: https://intuit.ru/studies/courses/17/17/info. (Дата обращения 26.08.2024).

Дополнительная литература

1. Зыков, С.В. Программирование. Объектно-ориентированный подход : учебник и практикум для студ. вузов, обуч. по инж.-техн. направ. и спец. / С.В. Зыков . ─ М. : Юрайт, 2022 . ─ 156 с. : ил. + табл. ─ (Высшее образование). – 3 экз.

Учебно-методическое обеспечение

Бизянов, Е.Е. Основы программирования. Часть 1 : лабораторный практикум / Е.Е. Бизянов, Л.В. Толстова. – Алчевск :ФГБОУ ВО "ДонГТУ", 2023) – 134 с. — URL: http://library.dstu.education/download.php?rec=103329.

3) Бизянов, Е.Е. Основы программирования. Часть 2 : лабораторный практикум / Е.Е. Бизянов, Л.В. Толстова. – Алчевск : ФГБОУ ВО "ДонГТУ", 2023) – 92 с. — URL: http://library.dstu.education/download.php?rec=132257.

7.2 Базы данных, электронно-библиотечные системы, информационно-справочные и поисковые системы

1. Научная библиотека ДонГТУ : официальный сайт.— Алчевск. — URL: library.dstu.education.— Текст : электронный.
2. Научно-техническая библиотека БГТУ им. Шухова : официальный сайт. — Белгород. — URL: http://ntb.bstu.ru/jirbis2/.— Текст : электронный.
3. Консультант студента : электронно-библиотечная система.— Москва. — URL: http://www.studentlibrary.ru/cgi-bin/mb4x.— Текст : электронный.
4. Университетская библиотека онлайн : электронно-библиотечная система.— URL: http://biblioclub.ru/index.php?page=main_ub_red.— Текст : электронный.
5. IPR BOOKS : электронно-библиотечная система.—Красногорск. — URL: http://www.iprbookshop.ru/. —Текст : электронный.

8 Материально-техническое обеспечение дисциплины

Материально-техническая база обеспечивает проведение всех видов деятельности в процессе обучения, соответствует требованиям ФГОС ВО.

Материально-техническое обеспечение представлено в таблице 7.

Таблица 7 – Материально-техническое обеспечение

Наименование оборудованных учебных кабинетов	Адрес (местоположение) учебных кабинетов
Специальные помещения: *Мультимедийная аудитория. (60 посадочных мест)*, оборудованная специализированной (учебной) мебелью (скамья учебная –20 шт., стол– 1 шт., доска аудиторная– 1 шт.), учебное ПК (монитор + системный блок), мультимедийная стойка с оборудованием – 1 шт., широкоформатный экран. Аудитории для проведения лекций	ауд. *207* корп. *4*
Компьютерный класс (11 посадочных мест), оборудованный учебной мебелью, компьютерами с неограниченным доступом к сети Интернет, включая доступ к ЭБС. Доска – 1 шт.	ауд. *211* корп. *4*

Лист согласования РПД

Разработал
И.о. заведующего кафедрой
интеллектуальных систем и
информационной безопасности
(должность) _____ (подпись) Е.Е. Бизянов (Ф.И.О.)

И.о. заведующего кафедрой
интеллектуальных систем и
информационной безопасности
(наименование кафедры) _____ (подпись) Е.Е. Бизянов (Ф.И.О.)

Протокол № 1 заседания кафедры от 27.08. 2024 г.

И.о. декана факультета
информационных технологий и
автоматизации производственных
процессов _____ (подпись) В.В Дьячкова (Ф.И.О.)

Согласовано

Председатель методической
комиссии по специальности 10.05.03 _____ (подпись) Е.Е. Бизянов (Ф.И.О.)
Информационная безопасность
автоматизированных систем

Начальник учебно-методического центра _____ (подпись О.А.Коваленко (Ф.И.О.)

Лист изменений и дополнений

Номер изменения, дата внесения изменения, номер страницы для внесения изменений	
ДО ВНЕСЕНИЯ ИЗМЕНЕНИЙ:	ПОСЛЕ ВНЕСЕНИЯ ИЗМЕНЕНИЙ:
Основание:	
Подпись лица, ответственного за внесение изменений	