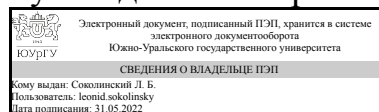


УТВЕРЖДАЮ:
Руководитель направления



Л. Б. Соколинский

РАБОЧАЯ ПРОГРАММА

дисциплины 1.О.11.03 Объектно-ориентированное программирование
для направления 02.03.02 Фундаментальная информатика и информационные
технологии

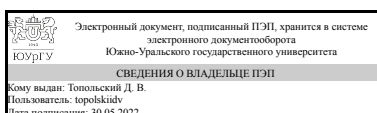
уровень Бакалавриат

форма обучения очная

кафедра-разработчик Электронные вычислительные машины

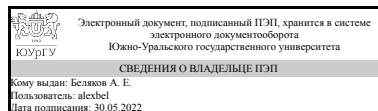
Рабочая программа составлена в соответствии с ФГОС ВО по направлению
подготовки 02.03.02 Фундаментальная информатика и информационные технологии,
утверждённым приказом Минобрнауки от 23.08.2017 № 808

Зав.кафедрой разработчика,
к.техн.н., доц.



Д. В. Топольский

Разработчик программы,
старший преподаватель



А. Е. Беляков

1. Цели и задачи дисциплины

Целью дисциплины является получение и закрепление базовых навыков применения объектно-ориентированной парадигмы проектирования и программирования компьютерных программ, знакомство в рамках данной парадигмы с обобщенным программированием и базовыми паттернами проектирования. Задачей дисциплины является освоение языка программирования C# и платформы .NET Framework.

Краткое содержание дисциплины

Дисциплина включает теоретическую и практическую части. На лекциях студенты изучают язык C#, на практике выполняют лабораторные работы (разработка компьютерных программ). Курс затрагивает следующие темы языка C#: понятие класса и объекта, инкапсуляция, наследование и полиморфизм в объектно-ориентированном программировании, примеры анализа, проектирования и разработки программных приложений для конкретной предметной области.

2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	Знает: функциональные возможности интегрированных сред разработки прикладного программного обеспечения на объектно-ориентированных языках программирования, возможности компиляторов программных проектов под различные операционные системы, наборы инструкций для системных утилит автоматической сборки программного обеспечения и установки программных пакетов объектно-ориентированных библиотек и фреймворков, средства мониторинга вычислительных ресурсов компьютерных программ Умеет: использовать возможности современных интегрированных сред разработки прикладного программного обеспечения на объектно-ориентированных языках программирования для разработки прикладных программ, использовать утилиты автоматической сборки и развертывания программ в операционных системах Имеет практический опыт: работы с современными интегрированными средами разработки прикладного программного обеспечения на объектно-ориентированных языках, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Unix и Windows
ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и	Знает: основные подходы к разработке прикладного программного обеспечения в рамках объектно-ориентированной парадигмы программирования на современном языке

<p>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</p>	<p>высокого уровня, основные принципы объектно-ориентированной парадигмы: абстрагирование, инкапсуляция, наследование, полиморфизм, базовые синтаксические конструкции объектно-ориентированного языка программирования: классы, поля, свойства, методы, выражения, события, методы обобщенного программирования, методы оценки сложности алгоритмов, функциональные возможности стандартной библиотеки языка и фреймворка Умеет: разрабатывать прикладное программное обеспечение в рамках объектно-ориентированной парадигмы на современном языке программирования высокого уровня с применением основных синтаксических конструкций и функциональных возможностей стандартной библиотеки языка и фреймворка Имеет практический опыт: разработки прикладного программного обеспечения, отладки, поиска и устранения ошибок программного кода, оценки сложности алгоритмов, использования возможностей стандартной библиотеки, сторонних библиотек программного кода и фреймворков</p>
--	---

3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
<p>1.О.11.01 Основы программирования, 1.О.21 Пакеты прикладных программ, 1.О.07 Информатика, 1.О.11.02 Программирование на языках высокого уровня, Учебная практика, технологическая (проектно-технологическая) практика (2 семестр)</p>	<p>ФД.02 Искусственный интеллект, 1.О.16 Технологии баз данных, 1.О.12 Операционные системы, 1.О.20 Компьютерные сети</p>

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин:

Дисциплина	Требования
<p>1.О.11.02 Программирование на языках высокого уровня</p>	<p>Знает: функциональные возможности интегрированных сред разработки прикладного и системного программного обеспечения на языках высокого уровня, особенности работы компиляторов и компоновщиков под различные операционные системы, наборы инструкций для системных утилит автоматической сборки программ, средства мониторинга вычислительных ресурсов компьютерных программ, основные подходы к разработке прикладных алгоритмов в рамках парадигмы структурного программирования на языке</p>

	<p>высокого уровня, базовые синтаксические конструкции языка программирования высокого уровня: операторы, выражения, блоки, ветвления, циклы; методы оценки сложности алгоритмов, функциональные возможности стандартной библиотеки языка высокого уровня, общие сведения об аппаратных и системных возможностях вычислительной техники для оптимизации программного обеспечения Умеет: использовать возможности современных интегрированных сред разработки прикладного и системного программного обеспечения на языках высокого уровня для разработки алгоритмов и программ, использовать утилиты автоматической сборки и развертывания программ в операционных системах, разрабатывать прикладное программное обеспечение в рамках парадигмы структурного программирования на языке программирования высокого уровня с применением основных синтаксических конструкций и функциональных возможностей стандартной библиотеки языка высокого уровня Имеет практический опыт: работы с современными интегрированными средами разработки прикладного и системного программного обеспечения на языках высокого уровня, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Unix и Windows, разработки прикладного программного обеспечения, отладки, поиска и устранения ошибок программного кода, оценки сложности алгоритмов, использования возможностей стандартной библиотеки, сторонних библиотек программного кода</p>
1.О.07 Информатика	<p>Знает: основные концепции, принципы, теории и факты, связанные с информатикой Умеет: применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой для проектирования и решения прикладных задач в различных областях, находить, агрегировать и использовать информацию связанную с профессиональной деятельностью Имеет практический опыт: владения навыками работы с технической документацией , управления информационными процессами (получения, хранения, передачи и обработки информации)</p>
1.О.21 Пакеты прикладных программ	<p>Знает: состав и функциональные возможности текстового редактора MS Word, издательской системы LaTeX, основные принципы компиляции программы и структуру стандартного компилятора Умеет: использовать возможности текстового редактора MS Word и издательской системы LaTeX, а также писать макросы на языке Visual Basic for Application,</p>

	разрабатывать обрабатывающий автомат на основе существующих синтаксических правил Имеет практический опыт: работы с современным текстовым редактором MS Word, издательской системой LaTeX, создания обрабатывающего автомата
1.О.11.01 Основы программирования	Знает: основные концепции структурного программирования, а также основные компоненты современной среды программирования, основные среды программирования, основные структуры представления данных и алгоритмы их обработки Умеет: применять методологию структурного программирования для решения задач, использовать современную среду программирования для создания и отладки программ, устанавливать и использовать основные возможности среды программирования, применять основные структуры данных и алгоритмы их обработки при решении поставленных задач, разрабатывать собственные алгоритмы и реализовывать их на языке высокого уровня Имеет практический опыт: решения простых задач с использованием парадигмы структурного программирования и современной среды программирования, установки и использования среды программирования, применения основных структур данных и алгоритмов их обработки при создании программ
Учебная практика, технологическая (проектно-технологическая) практика (2 семестр)	Знает: архитектуру программной системы заданного класса (определяется индивидуальным заданием), базовые алгоритмы и структуры данных Умеет: проектировать архитектуру разрабатываемой программной системы, использовать паттерны проектирования, разрабатывать и формулировать алгоритмы для решения поставленной задачи, описывать математические модели Имеет практический опыт: разработки моделей/алгоритмов/программ для решения поставленной задачи

4. Объём и виды учебной работы

Общая трудоемкость дисциплины составляет 4 з.е., 144 ч., 74,5 ч. контактной работы

Вид учебной работы	Всего часов	Распределение по семестрам в часах
		Номер семестра
		3
Общая трудоёмкость дисциплины	144	144
<i>Аудиторные занятия:</i>	64	64

Лекции (Л)	32	32
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	32	32
Лабораторные работы (ЛР)	0	0
<i>Самостоятельная работа (СРС)</i>	69,5	69,5
Разработка компьютерных программ	63,5	63,5
Подготовка к экзамену	6	6
Консультации и промежуточная аттестация	10,5	10,5
Вид контроля (зачет, диф.зачет, экзамен)	-	экзамен

5. Содержание дисциплины

№ раздела	Наименование разделов дисциплины	Объем аудиторных занятий по видам в часах			
		Всего	Л	ПЗ	ЛР
1	Введение в язык C# и платформу .NET	4	2	2	0
2	Сложность программных систем. Объектно-ориентированный подход	2	2	0	0
3	Реализация объектно-ориентированной парадигмы в C#	22	10	12	0
4	Обобщенное программирование	2	2	0	0
5	Потоковый ввод-вывод	10	4	6	0
6	Интерфейс обратного вызова	14	8	6	0
7	Обработка ошибок в C#	2	2	0	0
8	Многопоточность. Параллелизм. Асинхронность в C#	8	2	6	0

5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Определение языка C#. Компиляция. Управляемый модуль. Сборка. Возможности .NET	2
2	2	Основные понятия ООП (абстракция, модель, инкапсуляция, иерархия, полиморфизм)	2
3	3	Понятие класса, проектирование классов (атрибуты, методы, конструктор). Статические члены классов	2
4	3	Типы данных. Значащие и ссылочные типы. Передача параметров в методы	2
5	3	Строки. Массивы. Перегрузка операций	2
6	3	Наследование и полиморфизм. Абстрактный класс	2
7	3	Тип object. Интерфейсы	2
8	4	Обобщенные классы и методы. Стандартные объектные коллекции	2
9	5	Файловый ввод-вывод	2
10	5	Сериализация и десериализация объектов. Форматы XML и JSON	2
11	6	Делегаты. Лямбда-выражения	2
12	6	События	2
13	6	Итераторы	2
14	6	Язык интегрированных запросов LINQ	2
15	7	Многослойная архитектура. Класс Exception, операторы обработки ошибок	2
16	8	Паттерн Fork/Join. Библиотека TPL. PLINQ. Асинхронные операторы	2

5.2. Практические занятия, семинары

№ занятия	№ раздела	Наименование или краткое содержание практического занятия, семинара	Кол-во часов
1	1	Знакомство с компиляторами языка C# и IDE на разных ОС (Windows, Linux). Создание нового проекта и написание простейших программ в Visual Studio / Visual Studio Code	2
2	3	Понятие класса, структура класса (поля, свойства, методы), объекты в качестве параметров методов, возврат объектов из методов	6
3	3	Перегрузка операций в классах	6
4	5	Иерархия классов, наследование, полиморфизм. Абстрактные классы, виртуальные методы. Сериализация в XML/JSON. Обобщенные объектные коллекции	6
5	6	Делегаты и события. Механизм подписки на события. Класс web-клиент, http get-запрос	6
6	8	Web-клиент к API социальной сети. Windows-приложение. OAuth аутентификация	6

5.3. Лабораторные работы

Не предусмотрены

5.4. Самостоятельная работа студента

Выполнение СРС			
Подвид СРС	Список литературы (с указанием разделов, глав, страниц) / ссылка на ресурс	Семестр	Кол-во часов
Разработка компьютерных программ	Слайды лекций, методические указания к заданиям	3	63,5
Подготовка к экзамену	Слайды лекций, методические указания к заданиям	3	6

6. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации

Контроль качества освоения образовательной программы осуществляется в соответствии с Положением о балльно-рейтинговой системе оценивания результатов учебной деятельности обучающихся.

6.1. Контрольные мероприятия (КМ)

№ КМ	Се-мestr	Вид контроля	Название контрольного мероприятия	Вес	Макс. балл	Порядок начисления баллов	Учи-тыва-ется в ПА
1	3	Текущий контроль	ЛР №0	5	5	Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду. Шкала начисления дискретная, оформлена в виде набора критериев	экзамен

						<p>"рубрика" системы MOODLE. Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются. Критерии: 1. Наличие компилируемого без ошибок программного кода задания - 2 балла. 2. Понимание и комментирование функциональных блоков кода студентом - 2 балла. 3. Ответы на вопросы по заданию - 1 балл.</p>	
2	3	Текущий контроль	ЛР №1	10	10	<p>Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду. Шкала начисления дискретная, оформлена в виде набора критериев "рубрика" системы MOODLE. Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются. Критерии: 1. Наличие компилируемого без ошибок программного кода задания - 2 балла. 2. Понимание и комментирование функциональных блоков кода студентом - 6 баллов. 3. Ответы на вопросы по заданию - 2 балла.</p>	экзамен
3	3	Текущий контроль	ЛР №2	15	15	<p>Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду. Шкала начисления дискретная, оформлена в виде набора критериев "рубрика" системы MOODLE. Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются. Критерии: 1. Наличие компилируемого без ошибок программного кода задания - 5 баллов. 2. Понимание и комментирование функциональных блоков кода студентом - 6 баллов. 3. Ответы на вопросы по заданию - 4 баллов.</p>	экзамен
4	3	Текущий контроль	ЛР №3	20	20	<p>Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду. Шкала начисления дискретная, оформлена в виде набора критериев "рубрика" системы MOODLE. Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются. Критерии:</p>	экзамен

						1. Наличие компилируемого без ошибок программного кода задания - 6 баллов. 2. Понимание и комментирование функциональных блоков кода студентом - 10 баллов. 3. Ответы на вопросы по заданию - 4 баллов.	
5	3	Текущий контроль	ЛР №4	20	20	Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду. Шкала начисления дискретная, оформлена в виде набора критериев "рубрика" системы MOODLE. Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются. Критерии: 1. Наличие компилируемого без ошибок программного кода задания - 7 баллов. 2. Понимание и комментирование функциональных блоков кода студентом - 8 баллов. 3. Ответы на вопросы по заданию - 5 баллов.	экзамен
6	3	Текущий контроль	ЛР №5	10	10	Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду. Шкала начисления дискретная, оформлена в виде набора критериев "рубрика" системы MOODLE. Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются. Критерии: 1. Наличие компилируемого без ошибок программного кода задания - 2 балла. 2. Понимание и комментирование функциональных блоков кода студентом - 6 баллов. 3. Ответы на вопросы по заданию - 2 балла.	экзамен
7	3	Бонус	Бонусы	-	15	Баллы начисляются по результатам демонстрации бонусных программы и ответов на вопросы по исходному коду. Можно набрать максимум 15 баллов. Бонусные задачи оцениваются в зависимости от сложности от 5 до 15 баллов за задачу. 50% от максимума студент набирает за наличие компилируемого программного кода, 30% за корректное комментирование функциональных блоков кода, 20% за ответы на вопросы преподавателя.	экзамен
8	3	Промежуточная	Экзамен	-	100	По результатам ответов на вопросы устного экзамена.	экзамен

		аттестация				Первый теоретический вопрос - максимум 30 баллов, второй теоретический вопрос - максимум 30 баллов, практическая часть - максимум 40 баллов. За успешное выполнение ставится максимум баллов по заданию, если задание выполнено частично или с ошибками, то ставится от 0 до максимума по заданию пропорционально выполненному объему. Длительность экзамена - 2 академических часа.	
9	3	Текущий контроль	Промежуточный тест 1 (разделы 1, 2, 3)	5	5	Компьютерный тест по 1, 2 и 3 разделам курса, тест содержит 5 случайных равноценных вопросов. Время тестирования - 20 мин. Количество баллов за тест равно количеству правильных ответов студента.	экзамен
10	3	Текущий контроль	Промежуточный тест 2 (раздел 4)	5	5	Компьютерный тест по 4 разделу курса, тест содержит 5 случайных равноценных вопросов. Время тестирования - 20 мин. Количество баллов за тест равно количеству правильных ответов студента.	экзамен
11	3	Текущий контроль	Промежуточный тест 3 (раздел 5)	5	5	Компьютерный тест по 5 разделу курса, тест содержит 5 случайных равноценных вопросов. Время тестирования - 20 мин. Количество баллов за тест равно количеству правильных ответов студента.	экзамен
12	3	Текущий контроль	Промежуточный тест 4 (раздел 6)	5	5	Компьютерный тест по 6 разделу курса, тест содержит 5 случайных равноценных вопросов. Время тестирования - 20 мин. Количество баллов за тест равно количеству правильных ответов студента.	экзамен

6.2. Процедура проведения, критерии оценивания

Вид промежуточной аттестации	Процедура проведения	Критерии оценивания
экзамен	При оценивании результатов учебной деятельности обучающегося по дисциплине используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (Положение о БРС утверждено приказом ректора от 24.05.2019 г. № 179, в редакции приказа ректора от 10.03.2022 г. № 25-13/09). Оценка за дисциплину формируется на основе полученных оценок за контрольно-рейтинговые мероприятия текущего контроля. Отлично: Величина рейтинга обучающегося по дисциплине 85...100 %. Хорошо: Величина рейтинга обучающегося по дисциплине 75...84 %. Удовлетворительно: Величина рейтинга обучающегося по дисциплине 60...74 %. Неудовлетворительно:	В соответствии с пп. 2.5, 2.6 Положения

	<p>Величина рейтинга обучающегося по дисциплине 0...59 %.</p> <p>Если студент не согласен с оценкой, полученной по результатам текущего контроля, студент проходит мероприятие промежуточной аттестации в виде устного экзамена. Экзамен содержит два теоретических вопроса и практическую часть. На подготовку ответа по теории и выполнение практического задания дается 90 минут. В этом случае оценка за дисциплину рассчитывается на основе полученных оценок за контрольно-рейтинговые мероприятия текущего контроля и промежуточной аттестации. Фиксация результатов учебной деятельности по дисциплине проводится в день экзамена при личном присутствии студента.</p>	
--	--	--

6.3. Паспорт фонда оценочных средств

Компетенции	Результаты обучения	№ КМ															
		1	2	3	4	5	6	7	8	9	10	11	12				
ОПК-2	Знает: функциональные возможности интегрированных сред разработки прикладного программного обеспечения на объектно-ориентированных языках программирования, возможности компиляторов программных проектов под различные операционные системы, наборы инструкций для системных утилит автоматической сборки программного обеспечения и установки программных пакетов объектно-ориентированных библиотек и фреймворков, средства мониторинга вычислительных ресурсов компьютерных программ	+		+	+	+	+	+	+	+							
ОПК-2	Умеет: использовать возможности современных интегрированных сред разработки прикладного программного обеспечения на объектно-ориентированных языках программирования для разработки прикладных программ, использовать утилиты автоматической сборки и развертывания программ в операционных системах	+		+	+	+	+	+	+	+							
ОПК-2	Имеет практический опыт: работы с современными интегрированными средами разработки прикладного программного обеспечения на объектно-ориентированных языках, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Unix и Windows	+		+	+	+	+	+	+	+							
ОПК-3	Знает: основные подходы к разработке прикладного программного обеспечения в рамках объектно-ориентированной парадигмы программирования на современном языке высокого уровня, основные принципы объектно-ориентированной парадигмы: абстрагирование, инкапсуляция, наследование, полиморфизм, базовые синтаксические конструкции объектно-ориентированного языка программирования: классы, поля, свойства, методы, выражения, события, методы обобщенного программирования, методы оценки сложности алгоритмов, функциональные возможности стандартной библиотеки языка и фреймворка			+	+	+	+	+	+	+	+	+	+		+	+	
ОПК-3	Умеет: разрабатывать прикладное программное обеспечение в рамках объектно-ориентированной парадигмы на современном языке программирования высокого уровня с применением основных синтаксических конструкций и функциональных возможностей стандартной библиотеки языка и фреймворка			+	+	+	+	+	+	+	+	+		+	+		

ОПК-3	Имеет практический опыт: разработки прикладного программного обеспечения, отладки, поиска и устранения ошибок программного кода, оценки сложности алгоритмов, использования возможностей стандартной библиотеки, сторонних библиотек программного кода и фреймворков	<table border="1"> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table>																																										

Типовые контрольные задания по каждому мероприятию находятся в приложениях.

7. Учебно-методическое и информационное обеспечение дисциплины

Печатная учебно-методическая документация

а) основная литература:

Не предусмотрена

б) дополнительная литература:

Не предусмотрена

в) отечественные и зарубежные журналы по дисциплине, имеющиеся в библиотеке:

Не предусмотрены

г) методические указания для студентов по освоению дисциплины:

1. Программирование на языке С# (лекции)
2. Методические указания к заданию 2
3. Методические указания к заданию 3
4. Методические указания к заданию 1
5. Методические указания к заданию 4

из них: учебно-методическое обеспечение самостоятельной работы студента:

1. Программирование на языке С# (лекции)
2. Методические указания к заданию 2
3. Методические указания к заданию 3
4. Методические указания к заданию 1
5. Методические указания к заданию 4

Электронная учебно-методическая документация

№	Вид литературы	Наименование ресурса в электронной форме	Библиографическое описание
1	Основная литература	Электронно-библиотечная система издательства Лань	Суханов М.В. Бачурин И.В. Майров И.С. Основы Microsoft .NET Framework и языка программирования С#: учебное пособие https://e.lanbook.com/book/96543
2	Основная литература	Электронно-библиотечная система издательства Лань	Тюкачев Н.А. Хлебостроев В.Г. С# Алгоритмы и структуры данных https://e.lanbook.com/book/154117
3	Основная литература	Электронно-библиотечная система издательства Лань	Барков И.А. Объектно-ориентированное программирование: учебник https://e.lanbook.com/book/119661

4	Дополнительная литература	Электронно-библиотечная система издательства Лань	Голдштейн С. Зурбалева Д. Флатов И. Оптимизация приложений на платформе .NET https://e.lanbook.com/book/93266
---	---------------------------	---	---

Перечень используемого программного обеспечения:

1. -Microsoft Visual Studio (бессрочно)

Перечень используемых профессиональных баз данных и информационных справочных систем:

Нет

8. Материально-техническое обеспечение дисциплины

Вид занятий	№ ауд.	Основное оборудование, стенды, макеты, компьютерная техника, предустановленное программное обеспечение, используемое для различных видов занятий
Практические занятия и семинары	809 (36)	Компьютеры, ОС Windows, MS Visual Studio
Лекции	240 (36)	Компьютер, ОС Windows, MS Visual Studio, доска, проектор