Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Костровец Лариса Борисовна
Должность: директор
Дата подписания: 02.10.2026 00:12:43
Уникальный программный ключ:
6882606104c36dbde41c4ab93a65382136a292d6

Приложение 4
к образовательной программе

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.01.02.08 Объектно-ориентированное программирование
(индекс, наименование дисциплины в соответствии с учебным планом)

09.03.03 Прикладная информатика
(код, наименование направления подготовки/специальности)

Прикладная информатика в управлении корпоративными информационными системами
(наименование образовательной программы)

Очная форма обучения
(форма обучения)

Год набора – 2026
Донецк

Автор(ы)-составитель(и) РПД:

Лебезова Элла Михайловна, старший преподаватель кафедры информационных технологий

Заведующий кафедрой:

Брадул Наталья Валерьевна, канд. физ.-мат. наук, заведующий кафедрой информационных технологий

Рабочая программа дисциплины Б1.О.01.02.08 Объектно-ориентированное программирование одобрена на заседании кафедры информационных технологий администрирования факультета государственной службы и управления Донецкого филиала РАНХиГС.

Протокол № 7 от «05 » марта 2026 г.

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

2. Объем и место дисциплины в структуре образовательной программы

3. Содержание и структура дисциплины

4. Типы оценочных материалов, показатели и критерии их оценивания

5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам

6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине

7. Методические материалы по освоению дисциплины

8. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет»

9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Дисциплина Б1.О.01.02.08 Объектно-ориентированное программирование обеспечивает формирование у обучающихся следующих общепрофессиональных компетенций*:

ОТФ /ТФ и реквизиты ПС *(при наличии)* **	**Код компетенции ****	**Наименование Компетенции ****	**Код индикатора достижения компетенций ****	**Наименование индикатора достижения компетенций ****	**Образовательный результат ****
-	ОПК-7.	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.5.	Использует объектно-ориентированный подход для разработки программных приложений и решений	**Знает** принципы объектно-ориентированного программирования. **Умеет** применять объектно-ориентированный подход при разработке программных решений.

** Дисциплина может формировать компетенцию полностью или частично.*
***Должно соответствовать Приложению 1 к образовательной программе*

2. Объем и место дисциплины в структуре образовательной программы

Общий объем дисциплины:

3,00 з.е., 108 ак.час

Контактная работа обучающихся с преподавателем по видам учебных занятий: 47 ак. час на контактную работу с преподавателем, из них 14 ак. час на лекции и 24 ак. час на практические занятия. 61 ак. час на самостоятельную работу обучающихся.

Б1.О.01.02.08. Объектно-ориентированное программирование реализуется в 3-м семестре 2-го курса после изучения дисциплин:

- Информатика и программирование
- Алгоритмизация игровых задач
- Программирование на Python.

.

3. Содержание и структура дисциплины

3.1. Структура дисциплины

Очнаяформа обучения

№ п/п	Наименование тем и (или) разделов		Объем дисциплины, ак.час												Форма текущего контроля успеваемости, промежуточной аттестации
		ВСЕГО	Контактная работа обучающихся с преподавателем по видам учебных занятий									Самостоятельная работа			
			Период теоретического обучения						Период промежуточной аттестации (сессия)						
			Занятия лекционного типа		Занятия семинарского типа		ИК	КСР	КЭ	Каттэк	Контроль	СРкр	СРэк	СР	
			Л	ВЛ	ЛР	ПЗ									
РАЗДЕЛ 1. Объектно-ориентированное программирование на C#															
Тема 1	Введение в объектно-ориентированное программирование. Классы и объекты.	11	2	0	0	2	0	0	0	0	0	0	0	7	Контрольные вопросы, практические занятия, КР1
Тема 2	Основные принципы ООП.	15	2	0	0	4	0	0	0	0	0	0	0	9	Контрольные вопросы, практические

	Инкапсуляция в C#.													занятия, КР1
Тема 3	Основные принципы ООП. Наследование классов.	15	2	0	0	4	0	0	0	0	0	0	9	Контрольные вопросы, практические занятия, КР 1
Тема 4	Основные принципы ООП. Реализация полиморфизма	15	2	0	0	4	0	0	0	0	0	0	9	Контрольные вопросы, практические занятия, КР 1
РАЗДЕЛ 2. Построение программ на основе фундаментальных принципов ООП														
Тема 5	Интерфейсы и события в C #	15	2	0	0	4	0	0	0	0	0	0	9	Контрольные вопросы, практические занятия, КР 2
Тема 6	Программирование задач моделирования на основе ООП	16	2	0	0	4	0	0	0	0	0	0	10	Контрольные вопросы, практические занятия, КР 2
Тема 7	Шаблоны проектирования	12	2	0	0	2	0	0	0	0	0	0	8	Контрольные вопросы, практические занятия, КР 2
Промежуточная аттестация		9	0	0	0	0	0	0	0	9	0	0	0	0	Зачет с оценкой
Итого		**108**	**14**	**0**	**0**	**24**	**0**	**0**	**0**	**9**	**0**	**0**	**0**	**61**	

Используемые сокращения:

Л – лекции - занятия, предусматривающие преимущественную передачу учебной информации обучающимся педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях,).

ВЛ – видео лекции.
ЛР – лабораторные работы.
ПЗ – практические занятия (за исключением лабораторных работ).
ИК – индивидуальные консультации.
КСР – контроль самостоятельной работы
КЭ – консультации перед экзаменом
Каттэк – контактная работа на аттестацию в период экзаменационных сессий
Контроль - контактная работа на аттестацию в период экзаменационных сессий для заочной формы обучения
СРкр – самостоятельная работа на подготовку курсовой работы/ курсового проекта.
СРэк – самостоятельная работа на подготовку к экзамену.
СР – самостоятельная работа в семестре на подготовку к учебным занятиям.

3.2. Содержание дисциплины

РАЗДЕЛ 1. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ НА C#

Тема 1. Введение в объектно-ориентированное программирование. Классы и объекты. ОПК-7.5.

Понятие объектно-ориентированного подхода к разработке программного обеспечения. Отличия процедурного и объектно-ориентированного программирования. Основные элементы объектной модели: объект, класс, свойства, методы, поля. Среда разработки и особенности языка программирования Micrsft Visual Studi. Синтаксис описания классов и создания объектов в языке C#. Конструкторы классов, области видимости, спецификаторы доступа. Создание экземпляров классов и работа со ссылочными типами данных. Практическое применение классов и объектов при разработке простых программ.

Тема 2. Основные принципы ООП. Инкапсуляция в C#. ОПК-7.5.

Понятие инкапсуляции как механизма сокрытия внутренней реализации объекта. Использование модификаторов доступа private, public, prtected, internal. Организация безопасного доступа к данным класса. Свойства (prperties), методы доступа get и set. Автоматические свойства и контроль корректности данных. Применение инкапсуляции для повышения надежности и сопровождаемости программного кода. Разработка классов с управляемым доступом к данным в языке C#.

Тема 3. Основные принципы ООП. Наследование классов. ОПК-7.5.

Понятие наследования и иерархии классов. Базовые и производные классы. Повторное использование программного кода посредством наследования. Расширение функциональности существующих классов. Ключевые слова base и sealed. Особенности наследования в языке C#. Переопределение методов и использование конструкторов базового класса. Формирование иерархий классов для моделирования объектов предметной области.

Тема 4. Основные принципы ООП. Реализация полиморфизма. ОПК-7.5.

Понятие полиморфизма и его роль в объектно-ориентированном программировании. Статический и динамический полиморфизм. Перегрузка методов и операторов. Виртуальные методы, ключевые слова virtual, verride, abstract. Абстрактные классы и методы. Позднее связывание и работа с объектами через ссылки базового типа. Реализация гибких и расширяемых программных решений средствами полиморфизма в языке C#.

РАЗДЕЛ 2. ПОСТРОЕНИЕ ПРОГРАММ НА ОСНОВЕ ФУНДАМЕНТАЛЬНЫХ ПРИНЦИПОВ ООП

Тема 5. Интерфейсы и события в C#

Понятие интерфейса и его назначение в объектно-ориентированном программировании. Описание интерфейсов и их реализация в языке C#. Множественная реализация интерфейсов. Использование интерфейсов для обеспечения гибкости архитектуры программ. Делегаты и события. Механизм обработки событий в приложениях. Создание и обработка пользовательских событий. Практическое применение интерфейсов и событий при разработке программных систем.

Тема 6. Программирование задач моделирования на основе ООП

Применение объектно-ориентированного подхода для моделирования объектов и процессов предметной области. Анализ предметной области и выделение классов объектов. Построение взаимосвязей между объектами. Реализация программ моделирования с использованием классов, наследования, полиморфизма и интерфейсов. Разработка приложений, моделирующих реальные процессы и системы. Тестирование и отладка объектно-ориентированных программ.

Тема 7. Шаблоны проектирования

Понятие шаблонов проектирования и их роль в разработке программного обеспечения. Классификация шаблонов проектирования: порождающие, структурные и поведенческие шаблоны. Применение базовых шаблонов проектирования: Singletn, Factry Methd, bserver, Strategy и других. Повышение гибкости, расширяемости и сопровождаемости программных систем с использованием шаблонов проектирования. Практическая реализация шаблонов в языке C#.

4. Типы оценочных материалов, показатели и критерии оценивания

4.1. Оценочные материалы по дисциплине Б1.О.01.02.08. Объектно-ориентированное программирование входят в состав оценочных материалов по образовательной программе. Совокупность оценочных материалов по всем дисциплинам (модулям) образовательной программы составляет фонд оценочных средств (далее – ФОС). ФОС используется при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся с целью оценивания достижения обучающимися планируемых результатов обучения.

4.2. ФОС разработан как комплекс проверочных заданий различного типа и уровня сложности, включает критерии и шкалы оценивания, а также «ключи» правильных ответов. ФОС формируется как отдельный документ и хранится в электронном виде, доступ к ФОС предоставлен ограниченному кругу лиц.

4.3. Для самостоятельной работы обучающихся при подготовке к текущему контролю успеваемости и промежуточной аттестации в рабочих программах дисциплин размещены типовые проверочные задания, которые можно условно разделить на задания закрытого, комбинированного и открытого типов.

Задания закрытого типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных.

Задания комбинированного типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных и обосновать свой выбор.

Задания открытого типа – это задания, в которых на каждый вопрос должен быть предложен развернутый обоснованный ответ.

В зависимости от типа задания рекомендованы определенная последовательность выполнения и система оценивания выполнения заданий.

4.4. Типы заданий, сценарии выполнения, критерии оценивания

ТИП ЗАДАНИЯ	ИНСТРУКЦИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	КРИТЕРИИ ОЦЕНИВАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных	Прочитайте текст, выберите правильный ответ	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В).	Ответ считается верным, если правильно указана цифра или буква
Задание закрытого типа на установление соответствия	Прочитайте текст и установите соответствие	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).	Ответ считается верным, если правильно указаны цифры или буквы

Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных	Прочитайте текст, выберите правильные ответы	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).	Ответ считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)
Задание закрытого типа на установление последовательнос ти	Прочитайте текст и установите последовательность	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты ответа. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135).	Ответ считается верным, если правильно указана вся последовательность цифр

Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора	Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа. 5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования).	Ответ считается верным, если правильно указана цифра или буква и приведены корректные аргументы, используемые при выборе ответа
Задание открытого типа с развернутым ответом	Прочитайте текст и запишите развернутый обоснованный ответ	1. Внимательно прочитать текст задания и понять суть вопроса. 2. Продумать логику и полноту ответа. 3. Записать ответ, используя четкие компактные формулировки. 4. В случае расчетной задачи, записать решение и ответ	Ответ считается верным: 1. Отсутствие фактических ошибок. 2. Раскрытие объема используемых понятий (полнота ответа). 3. Обоснованность ответа (наличие аргументов). 4. Логическая последовательность излагаемого материала.

4.5. Общая шкала оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся с применением БРС

Оценка по шкале ECTS	**Сумма баллов за все виды учебной деятельности**	**Оценка по государственной шкале**	**Определение**
A	90 – 100	«Отлично»	отличное выполнение с незначительным количеством неточностей
B	80 – 89	«Хорошо»	в целом правильно выполненная работа с незначительным количеством ошибок (до 10%)
C	75 – 79		в целом правильно выполненная работа с незначительным количеством ошибок (до 15%)
D	70 – 74	«Удовлетворительно»	неплохо, но со значительным количеством недостатков
E	60 – 69		выполнение удовлетворяет минимальные критерии
FX	35 – 59	«Не удовлетворительно»	с возможностью повторной сдачи
F	0 – 34		с обязательным повторным изучением дисциплины (выставляется комиссией)

Соотношение баллов за текущий контроль успеваемости и промежуточную аттестацию, а также повторную промежуточную аттестацию:

Максимальная сумма баллов за текущий контроль успеваемости	Максимальная сумма баллов за промежуточную аттестацию	Максимальная итоговая балльная оценка	Максимальная сумма баллов за повторную промежуточную аттестацию
100 баллов	100 баллов	100 баллов	100 баллов

5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам

5.1. В ходе реализации дисциплины Б1.О.01.02.08 Объектно-ориентированное программирование используются следующие формы

текущего контроля успеваемости обучающихся (в том числе, задания к контрольным точкам):

Контрольные вопросы для проведения опроса, задания открытого типа на практических занятиях, контрольные задания

Таблица 5.1.

Распределение баллов по видам учебной деятельности (БРС)

Раздел/Темы	Формы текущего контроля		КЗР
	УО	ПЗ	
Р-1. / Т-1	3	7	15
Р-1. / Т-2	3	7	
Р-1. / Т-3	3	7	
Р-1. / Т-4	3	7	15
Р-2. / Т-5	3	7	
Р-2. / Т-6	3	7	
Р-2. / Т-7	3	7	
Итого: 100 б	18	7	30

УО – устный опрос;
ТЗ – тестовое задание;
КЗ – контрольные задания;
ПЗ – практическое занятие;
Д – доклад;
КЗР – контрольные работы по разделу.

Критерии оценивания опроса:

Балы	Описание критерия
3	Обучающийся полно излагает материал (отвечает на вопрос), дает правильное определение основных понятий; обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; излагает материал последовательно и правильно с точки зрения норм литературного языка.
2	Обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.
1	Обучающийся обнаруживает знание и понимание основных положений данной темы, но излагает материал неполно и допускает неточности в определении понятий или формулировке правил; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
0	Обучающийся обнаруживает незнание вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал.

0* - в журнал академической группы не выставляется

Критерии оценивания практических занятий:

Балы	Описание критерия	
3	Свыше 90% правильных ответов.	Обучающийся демонстрирует глубокое познание в освоенном материале.
2	Свыше 70% правильных ответов.	Обучающимся материал освоен полностью, без существенных ошибок.
1	Реализовано более 50% поставленных задач	Обучающимся материал освоен не полностью, имеются значительные пробелы в знаниях.
0	Реализовано менее 30% поставленных задач.	Обучающимся материал не освоен, знания обучающегося ниже базового уровня.

0* - в журнал академической группы не выставляется

Критерии оценивания контрольных заданий:

Балы	Описание критерия
12-15	Обучающимся задание выполнено без ошибок и в полном объеме.
8-11	Обучающимся в целом задание выполнено, имеются отдельные неточности или недостаточно полные ответы, не содержащие ошибок.
5-7	Обучающимся допущены отдельные ошибки при выполнении задания
0-4	У обучающегося отсутствуют ответы на большинство вопросов задачи, задание не выполнено или выполнено не верно.

0* - в журнал академической группы не выставляется

5.2. Типовые оценочные материалы для текущего контроля успеваемости обучающихся (вне контрольных работ):

РАЗДЕЛ 1. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ НА C#

Тема 1. Введение в объектно-ориентированное программирование. Классы и объекты

Контрольные вопросы:

1. Что такое объектно-ориентированное программирование?
2. В чем отличие процедурного программирования от ООП?
3. Что такое класс и объект?
4. Какие элементы включает класс?
5. Что такое поле, свойство и метод?
6. Для чего используются конструкторы?
7. Какие существуют спецификаторы доступа в C#?
8. Что представляет собой экземпляр класса?
9. Что такое ссылочный тип данных?
10. Как создаются объекты в C#?

Практические занятия:

Создать класс Student, содержащий следующие члены класса:

1. Поля:

– фамилия студента;
– номер группы;
– средний балл.

2. Конструкторы:

– конструктор без параметров;
– конструктор с параметрами.

3. Методы:

– вывод информации о студенте на экран;
– изменение среднего балла;
– определение, является ли студент отличником.

4. Свойства:

– для чтения и записи данных студента;
– для получения статуса успеваемости.

5. Создать несколько объектов класса Student и продемонстрировать работу методов и свойств.

Тема 2. Основные принципы ООП. Инкапсуляция в C#

Контрольные вопросы:

1. Что такое инкапсуляция?
2. Для чего используются модификаторы доступа?
3. Чем отличаются private и public?
4. Что такое свойства класса?
5. Для чего применяются аксессоры get и set?
6. Что такое автоматические свойства?
7. Как обеспечивается защита данных объекта?
8. В чем преимущества инкапсуляции?
9. Что такое внутреннее состояние объекта?
10. Почему важно ограничивать прямой доступ к полям класса?

Практические занятия:

Создать класс BankAccount, содержащий следующие члены класса:

1. Поля:

– номер счета;
– владелец счета;
– баланс.

2. Конструкторы:

– без параметров;
– с параметрами.

3. Методы:

– пополнение счета;
– снятие средств;
– вывод информации о счете.

4. Свойства:
– для чтения номера счета;
– для чтения и изменения владельца;
– для контролируемого изменения баланса.
5. Реализовать проверку корректности операций снятия средств.
6. Продемонстрировать работу класса.

Тема 3. Основные принципы ООП. Наследование классов

Контрольные вопросы:
1. Что такое наследование?
2. Что называется базовым классом?
3. Что называется производным классом?
4. Для чего используется наследование?
5. Что такое иерархия классов?
6. Как вызывается конструктор базового класса?
7. Для чего используется ключевое слово base?
8. Что означает sealed-класс?
9. Какие преимущества дает наследование?
10. Какие ограничения существуют при наследовании?

Практические занятия:
Создать базовый класс Transport и производные классы Car и Bicycle.
1. Класс Transport должен содержать:
– название транспортного средства;
– максимальную скорость;
– метод вывода информации.
2. Класс Car дополнительно должен содержать:
– тип топлива;
– количество дверей.
3. Класс Bicycle дополнительно должен содержать:
– тип велосипеда;
– наличие корзины.
4. Переопределить метод вывода информации для каждого производного класса.
5. Создать массив объектов базового класса и продемонстрировать работу программы.

Тема 4. Основные принципы ООП. Реализация полиморфизма

Контрольные вопросы:
1. Что такое полиморфизм?
2. Какие виды полиморфизма существуют?
3. Что такое перегрузка методов?
4. Что такое переопределение методов?

5. Для чего используются virtual и override?
6. Что такое абстрактный класс?
7. Чем абстрактный класс отличается от обычного?
8. Что такое позднее связывание?
9. В чем преимущества полиморфизма?
10. Как реализуется динамический полиморфизм в C#?

Практические занятия:

Создать абстрактный класс Shape и производные классы Circle, Rectangle и Triangle.

1. Класс Shape должен содержать:

– абстрактный метод вычисления площади;
– виртуальный метод вывода информации.

2. Производные классы должны:

– реализовать вычисление площади;
– переопределить метод вывода информации.

3. Создать коллекцию объектов различных фигур.
4. Организовать вывод площади всех фигур через общий интерфейс работы с базовым классом.
5. Продемонстрировать работу полиморфизма.

РАЗДЕЛ 2. ПОСТРОЕНИЕ ПРОГРАММ НА ОСНОВЕ ФУНДАМЕНТАЛЬНЫХ ПРИНЦИПОВ ООП

Тема 5. Интерфейсы и события в C#

Контрольные вопросы:

1. Что такое интерфейс?
2. Чем интерфейс отличается от абстрактного класса?
3. Может ли класс реализовывать несколько интерфейсов?
4. Для чего используются интерфейсы?
5. Что такое делегат?
6. Что такое событие?
7. Как объявляются события в C#?
8. Как происходит подписка на событие?
9. Что такое обработчик события?
10. Где применяются события?

Практические занятия:

Создать интерфейс IPlayable и классы AudioPlayer и VideoPlayer.

1. Интерфейс должен содержать методы:

– Play();
– Pause();
– Stop().

2. Классы AudioPlayer и VideoPlayer должны реализовать интерфейс.
3. Реализовать событие начала воспроизведения.
4. Организовать вывод сообщения при возникновении события.
5. Продемонстрировать работу интерфейсов и событий.

Тема 6. Программирование задач моделирования на основе ООП

Контрольные вопросы:

1. Что понимается под моделированием в программировании?
2. Какие этапы включает моделирование предметной области?
3. Как определить классы объектов?
4. Что такое ассоциация между объектами?
5. Для чего используются интерфейсы при моделировании?
6. Как применяется наследование в моделировании?
7. Что такое композиция объектов?
8. Как организуется взаимодействие объектов?
9. Какие преимущества дает объектное моделирование?
10. Как выполняется тестирование объектной модели?

Практические занятия:

Разработать программу моделирования работы библиотеки.

1. Создать классы:

– Book;
– Reader;
– Library.

2. Класс Book должен содержать:

– название книги;
– автора;
– год издания.

3. Класс Reader должен содержать:

– ФИО читателя;
– номер читательского билета.

4. Класс Library должен обеспечивать:

– добавление книг;
– выдачу книг читателям;
– возврат книг;
– поиск книг.

5. Реализовать хранение объектов в коллекциях.
6. Продемонстрировать работу программы.

Тема 7. Шаблоны проектирования

Контрольные вопросы:

1. Что такое шаблон проектирования?

2. Для чего применяются шаблоны проектирования?
3. Какие группы шаблонов существуют?
4. Что такое порождающие шаблоны?
5. Что такое структурные шаблоны?
6. Что такое поведенческие шаблоны?
7. В чем назначение шаблона Singleton?
8. Для чего используется Factory Method?
9. Что такое шаблон Observer?
10. Какие преимущества дают шаблоны проектирования?

Практические занятия:

Реализовать программу управления настройками приложения с использованием шаблона Singleton.

1. Создать класс AppConfig:

– хранение настроек приложения;
– закрытый конструктор;
– статический объект класса.

2. Реализовать:

– получение единственного экземпляра класса;
– методы изменения и отображения настроек.

3. Проверить невозможность создания нескольких объектов класса.
4. Дополнительно реализовать шаблон Factory Method для создания различных типов отчетов.
5. Продемонстрировать работу шаблонов проектирования в программе.

5.3. Один или несколько тематических блоков дисциплины завершаются контрольной работой по разделу (далее – КР). Текущий контроль успеваемости по дисциплине предусматривает не менее 2 (двух) и не более 10 (десяти) КР в течение периода освоения дисциплины.

Максимальное количество баллов за любой тип работ в рамках КР составляет 100 (сто) баллов.

Распределение весовых коэффициентов по КР в рамках текущего контроля успеваемости по дисциплине и формулы расчета:

Наименование контрольной работы	Максимальное количество баллов за работу в рамках КР, которое может набрать студент	Коэффициент веса контрольной работы	Результат контрольной работы, участвующий в формировании итоговой балльной оценки по дисциплине

КР 1	100	0,15	15
КР 2	100	0,15	15
Итого:	x	0,30	30

Формула расчета результата контрольной работы:

Результат контрольной работы = Количество баллов за работу в рамках КР X Коэффициент веса контрольной работы.

5.4. Формы текущего контроля успеваемости обучающихся в рамках КР и типовые оценочные материалы:

КР-1

Раздел 1. Объектно-ориентированное программирование на C#

Теоретические вопросы:

1. Что такое класс и объект в объектно-ориентированном программировании?
2. Для чего используются конструкторы?
3. Что такое инкапсуляция?
4. В чем заключается наследование классов?
5. Для чего применяется полиморфизм?

Практическое задание:

Создать программу для работы с информацией о сотрудниках предприятия.

1. Создать базовый класс Employee, содержащий:

– фамилию сотрудника;
– должность;
– размер заработной платы.

2. Реализовать:

– конструкторы;
– метод вывода информации;
– свойства для доступа к данным.

3. Создать производный класс Manager, содержащий:

– количество подчиненных.

4. Переопределить метод вывода информации в производном классе.
5. Создать несколько объектов классов и продемонстрировать работу программы.

КР-2

Раздел 2. Построение программ на основе фундаментальных принципов ООП

Теоретические вопросы:

1. Что такое интерфейс?
2. Для чего используются события в C#?
3. Что понимается под моделированием предметной области?
4. Что такое шаблон проектирования?
5. В чем назначение шаблона Singleton?

Практическое задание:

Разработать программу учета товаров на складе.

1. Создать интерфейс IInfo, содержащий метод вывода информации.
2. Создать класс Product, содержащий:

– наименование товара;
– количество;
– цену.

3. Реализовать интерфейс IInfo в классе Product.
4. Создать класс Warehouse, обеспечивающий:

– добавление товаров;
– поиск товара;
– вывод списка товаров.

5. Реализовать событие добавления нового товара на склад.
6. Продемонстрировать работу программы.

6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине

6.1. Промежуточная аттестация по дисциплине «Объектно-ориентированное программирование» проводится в форме зачёта с оценкой в третьем семестре в письменной форме. Обучающийся получает три теоретических вопроса и одно практическое задание.

6.2. Типовые оценочные материалы промежуточной аттестации.

Вопросы к зачету с оценкой:

1. Понятие объектно-ориентированного программирования.
2. Основные отличия объектно-ориентированного и процедурного программирования.
3. Понятие класса и объекта.
4. Поля, свойства и методы класса.
5. Конструкторы классов и их назначение.
6. Спецификаторы доступа в C#.
7. Инкапсуляция как принцип ООП.
8. Свойства и методы доступа get и set.
9. Автоматические свойства в C#.
10. Наследование классов и его назначение.
11. Базовые и производные классы.
12. Использование ключевого слова base.
13. Переопределение методов.
14. Полиморфизм и его виды.
15. Виртуальные и абстрактные методы.
16. Абстрактные классы в C#.
17. Перегрузка методов и операторов.
18. Интерфейсы и их назначение.
19. Реализация интерфейсов в C#.
20. Отличия интерфейсов от абстрактных классов.
21. Делегаты в C#.
22. События и механизм их обработки.
23. Коллекции объектов в C#.
24. Основы объектного моделирования.
25. Ассоциация и композиция объектов.
26. Применение ООП при разработке программ.
27. Понятие шаблонов проектирования.
28. Классификация шаблонов проектирования.
29. Шаблон Singleton.
30. Шаблон Factory Method.
31. Шаблон Observer.

32. Преимущества использования шаблонов проектирования.
33. Обработка исключений в C#.
34. Пространства имен и организация программного кода.
35. Практическое применение объектно-ориентированного программирования при разработке программных систем.

Пример практического задания

Разработать объектно-ориентированное приложение «Учет студентов учебной группы».

1. Создать класс Student, содержащий:
– фамилию и имя студента;
– номер группы;
– средний балл.
2. Реализовать:
– конструкторы класса;
– свойства для доступа к данным;
– метод вывода информации о студенте;
– метод определения статуса успеваемости студента.
3. Создать производный класс GraduateStudent, содержащий:
– тему выпускной квалификационной работы.
4. Переопределить метод вывода информации в производном классе.
5. Создать интерфейс IPrintable с методом Print() и реализовать его в созданных классах.
6. Организовать хранение объектов в коллекции.
7. Реализовать поиск студента по фамилии.
8. Продемонстрировать работу программы:
– создание объектов;
– вывод информации;
– поиск объектов;
– использование наследования и полиморфизма.

6.3. Критерии и шкала оценивания на основе БРС.

Соответствие государственной шкалы оценивания академической успеваемости и шкалы ECTS при зачёте

Оценка по шкале ECTS	**Сумма баллов за все виды учебной деятельности**	**Оценка по государственной шкале**	**Определение**
A	90 – 100	«Отлично»	отличное выполнение с незначительным количеством неточностей
B	80 – 89	«Хорошо»	в целом правильно выполненная работа с незначительным количеством ошибок (до 10%)
C	75 – 79		в целом правильно выполненная работа с незначительным количеством ошибок (до 15%)
D	70 – 74	«Удовлетворительно»	неплохо, но со значительным количеством недостатков
E	60 – 69		выполнение удовлетворяет минимальные критерии
FX	35 – 59	«Не удовлетворительно»	с возможностью повторной сдачи
F	0 – 34		с обязательным повторным изучением дисциплины (выставляется комиссией)

6.4. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий

Компьютер с операционной системой Windws с устойчивым Интернет-соединением, программные продукты с открытой лицензией: Visual Studio.

7. Методические материалы по освоению дисциплины

Получение углубленных знаний по изучаемой дисциплине достигается за счет дополнительных часов к аудиторной работе самостоятельной работы студентов. Выделяемые часы целесообразно использовать для знакомства с дополнительной научной литературой по проблематике дисциплины, анализа научных концепций и современных подходов к осмыслению рассматриваемых проблем. К самостоятельному виду работы студентов относится работа в библиотеках, в электронных поисковых системах и т.п. по сбору материалов, необходимых для проведения практических занятий или выполнения конкретных заданий преподавателя по изучаемым темам. Студенты могут установить диалог с преподавателем, получать консультации по выполнению заданий. В качестве оценочных средств на протяжении семестра используются практические задания.

Обучение по дисциплине «Объектно-ориентированное программирование» предполагает изучение курса на аудиторных занятиях (лекции, практические занятия) и самостоятельную работу студентов. Практические занятия дисциплины предполагают их проведение в различных формах с целью выявления полученных знаний, умений, навыков и компетенций с проведением контрольных мероприятий. С целью обеспечения успешного обучения студент должен готовиться к лекции, поскольку она является важнейшей формой организации учебного процесса, поскольку:

– знакомит с новым учебным материалом;
– разъясняет учебные элементы, трудные для понимания;
– систематизирует учебный материал;
– ориентирует в учебном процессе.

Работа обучающегося на лекции:

Слушание и запись лекций – сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность обучающегося. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим обучающимся.

Подготовка к практическим занятиям:

Подготовку к каждому практическому занятию каждый обучающийся должен начать с ознакомления с планом, который отражает содержание предложенной темы. Тщательное продумывание и изучение вопросов плана основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованную к данной теме. Если программой дисциплины предусмотрено выполнение практического задания, то его необходимо выполнить с учетом предложенной инструкции. Все новые понятия по изучаемой теме необходимо внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности обучающегося свободно ответить на теоретические вопросы практического занятия, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий и контрольных работ.

Структура практического занятия:

В зависимости от содержания и количества отведенного времени на изучение каждой темы может практическое занятие состоять из четырех-пяти частей:

1. Устный опрос.

2. Обсуждение теоретических вопросов, определенных программой дисциплины.

3. Выполнение практических заданий с последующим разбором полученных результатов или обсуждение практического задания, выполненного дома.

4. Подведение итогов занятия.

Работа с литературными источниками:

В процессе подготовки к практическим занятиям, обучающимся необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем, что позволяет обучающимся проявить свою индивидуальность в рамках выступления на занятиях, выявить широкий спектр мнений по изучаемой проблеме.

8. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет

8.1. Основная литература

1. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# / Л. А. Залогова. — 4-е изд., стер. — Санкт-Петербург : Лань, 2023. — 192 с. — ISBN 978-5-507-48276-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/345992 (дата обращения: 13.05.2026). — Режим доступа: для авториз. пользователей.

2. Вафин, Р. Р. Объектно-ориентированное программирование на C#.NET : учебно-методическое пособие / Р. Р. Вафин. — Казань : КНИТУ-КАИ, 2020. — 96 с. — ISBN 978-5-7579-2431-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/264920 (дата обращения: 13.05.2026). — Режим доступа: для авториз. пользователей.

8.2. Дополнительная литература

3. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# : учебное пособие для СПО / Л. А. Залогова. — 4-е изд., стер. — Санкт-Петербург : Лань, 2025. — 192 с. — ISBN 978-5-507-51145-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/505852 (дата обращения: 13.05.2026). — Режим доступа: для авториз. пользователей.

8.3.Нормативные правовые документы и иная правовая информация

1. Конституция Российской Федерации. – Текст : электронный // Сайт Президента Российской Федерации. – URL: http://www.kremlin.ru/acts/cnstitutin

8.4 Интернет-ресурсы

1. Информационно-правовой портал ГАРАНТ.РУ. – URL: https://www.garant.ru/

2. Научная электронная библиотека eLIBRARY.RU. – URL: https://elibrary.ru/

3. Научная электронная библиотека «КиберЛенинка». – URL: https://cyberleninka.ru

4. Электронно-библиотечная система «Лань». – URL: http://e.lanbk.cm

5. База знаний по C# – URL: https://metanit.com/sharp/tutorial/

6. Документация по C# – https://learn.microsoft.com/ru-ru/dotnet/csharp/language-reference/

9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

Материально-техническое обеспечение дисциплины включает в себя:

- лекционные аудитории, оборудованные видеопроекционным оборудованием для презентаций, средствами звуковоспроизведения, экраном;
- помещения для проведения практических занятий, оборудованные учебной мебелью.

Дисциплина поддержана соответствующими программными продуктами с открытой лицензией: Visual Studio.

Вуз обеспечивает каждого обучающегося рабочим местом в компьютерном классе в соответствии с объемом изучаемых дисциплин, обеспечивает выход в сеть Интернет.

Помещения для самостоятельной работы обучающихся включают следующую оснащенность: столы аудиторные, стулья, доски аудиторные, компьютеры с подключением к локальной сети института (для компьютерных аудиторий) и Интернет. Для изучения учебной дисциплины используются автоматизированная библиотечная информационная система и электронные библиотечные системы.