МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И.Носова»

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ НА C#

Направление подготовки (специальность)
09.03.03 Прикладная информатика

Направленность (профиль/специализация) программы
Разработка компьютерных игр и AR/VR-приложений (виртуальной/дополненной реальности)

Уровень высшего образования - бакалавриат

Форма обучения
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Бизнес-информатики и информационных технологий
Курс	2
Семестр	3

Магнитогорск
2026 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

Рабочая программа рассмотрена и одобрена на заседании кафедры Бизнес-информатики и информационных технологий
22.01.2026, протокол № 5

Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа одобрена методической комиссией ИЭиАС
03.02.2026 г. протокол № 3

Председатель _____ В.Р. Храмшин

Рабочая программа составлена:
доцент кафедры кафедры БИиИТ, канд. пед. наук
_____ И.В.Гаврилова

Рецензент:
главный специалист службы бизнес-анализа,
КОНСОМ ГРУПП, канд.техн. наук _____ В.А. Ошурков

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028учебном году на заседании кафедры Бизнес-информатики и

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029учебном году на заседании кафедры Бизнес-информатики и

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2029 - 2030учебном году на заседании кафедры Бизнес-информатики и

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2030 - 2031учебном году на заседании кафедры Бизнес-информатики и

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ Г.Н. Чусавитина

1 Цели освоения дисциплины (модуля)

Основной целью дисциплины является формирование понимания идеологии и ключевых аспектов объектно-ориентированного программирования (ООП) на языке C#, достаточного для практического использования в процессе дальнейшего обучения и в профессиональной сфере.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Объектно-ориентированное программирование на C# входит в часть учебного плана, формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения),сформированные в результате изучения дисциплин/ практик:

Программирование

Информатика

Знания (умения, владения), полученные при изучении данной дисциплины, будут необходимы для изучения дисциплин/практик:

Геймдизайн и основы игровой логики

Проектирование информационных систем

Разработка компьютерных игр и мультимедийных приложений

Разработка приложений виртуальной и дополненной реальности

Разработка UI/UX

Производственная – преддипломная практика

Подготовка к сдаче и сдача государственного экзамена

Мультиплеерные игры

Выполнение и защита выпускной квалификационной работы

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Объектно-ориентированное программирование на C#» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-1 Способен разрабатывать компьютерные игры, AR/VR -приложения	
ПК-1.1	Проводит обследование организаций, выявляет информационные потребности пользователей, анализирует и формирует требования к мультимедийным приложениям
ПК-1.2	Проектирует мультимедийные приложения (компьютерные игры и приложения виртуальной/дополненной реальности)
ПК-1.3	Участвует в реализации проектов по созданию мультимедийных приложений под различные платформы и устройства
ПК-1.4	Осуществляет тестирование мультимедийных приложений
ПК-FS Разработка виртуальной и дополненной реальности	
ПК-FS.1	Знает основные понятия и концепции в области цифровых реальностей, основные сенсомоторные и психологические характеристики человека, включаемые в системы цифровых реальностей, виды и классификации систем цифровых реальностей
ПК-FS.2	Знает современные программные и аппаратные средства ихреализации, проектные и технические процессы их создания,основные стандарты и методы оценивания пригодности при разработке систем цифровых реальностей, области и примеры их использования

ПК-FS.3	Умеет разрабатывать требования и архитектуру приложений на базе систем цифровых реальностей, выбирать технологии и инструменты их реализации
ПК-FS.4	Умеет разрабатывать методы, модели, алгоритмы и программы приложений на базе систем цифровых реальностей, оценивать пригодность их использования, реализовывать проектные и технические процессы их создания
ПК-FS.5	Владеет навыками разработки и оценки приложений на базе систем цифровых реальностей с использованием современных аппаратных и программных средств
ПК-FS.6	Владеет навыками применения стандартов при составлении технической документации на разработку, испытание и использование приложений на базе систем цифровых реальностей

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 зачетных единиц 144 акад. часов, в томчисле:
– контактная работа – 91,9 акад. часов:
– аудиторная – 90 акад. часов;
– внеаудиторная – 1,9 акад. часов;
– самостоятельная работа – 52,1 акад. часов;
– в форме практической подготовки – 0 акад. час;

Форма аттестации - зачет

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельнаяработа студента	Видсамостоятельной работы	Форма текущегоконтроляуспеваемости и промежуточнойаттестации	Кодкомпетенции
		Лек.	лаб. зан.	практ.зан.				
1. Основы языка программирования								
1.1 Система типов данных C#. Объявление переменных. Преобразование типов. Консольный ввод и вывод	3	2	2/2И		3	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.2, ПК-1.3, ПК-1.4,ПК-FS.2, ПК-FS.3, ПК-FS.4, ПК-FS.5
1.2 Выражения и операциив C#. Операторы языка: простые и составные, операторы выбора, цикла, перехода.		2	2/2И		3	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.2, ПК-1.3, ПК-1.4,ПК-FS.2, ПК-FS.3, ПК-FS.4, ПК-FS.5
1.3 Работа с наборами данных: массивы (одномерные, многомерные, неровные), перечисления и структуры, коллекции (списки,очереди		4	6/6И		3	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.2, ПК-1.3, ПК-1.4,ПК-FS.2, ПК-FS.3, ПК-FS.4, ПК-FS.5
1.4 Классы: объявление свойств и методов. Интерфейсы. Делегаты и асинхронные делегаты		6	6/2И		3	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.2, ПК-1.3, ПК-1.4,ПК-FS.2, ПК-FS.3, ПК-FS.4, ПК-FS.5

1.5 Наследование и полиморфизм		4	4/4И		3	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.2, ПК-1.3, ПК-1.4, ПК-FS.2, ПК-FS.3, ПК-FS.4, ПК-FS.5, ПК-FS.6
1.6 Обработка исключительных ситуаций	3	2	2		3	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.2, ПК-1.3, ПК-1.4,ПК-FS.2, ПК-FS.3, ПК-FS.4, ПК-FS.5
Итого по разделу		20	22/16И		18			
2. Разработка приложений на основе объектно-ориентированного подхода								
2.1 Объектно-ориентированное проектирование программных систем. Основы UML: диаграммы классов, объектов, компонентов, пакетов, деятельности, вариантов использования, коммуникации, последовательности, сотрудничества, синхронизации.	3	2	6		4	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.2, ПК-1.3, ПК-1.4,ПК-FS.2, ПК-FS.3, ПК-FS.4, ПК-FS.5
2.2 Среды визуальной разработки. MS VisualStudio, Project Rider, Eclipse, Visual Studio Code, SharpDevelop IDE. Структура проекта, пространства имён,библиотеки.		2/2И	2		3	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-FS.2
2.3 Основы разработки графического пользовательского интерфейса: формы, элементы управления, диалоги.		2/2И	6		5	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.3, ПК-1.4, ПК-FS.3,ПК-FS.4, ПК-FS.5
2.4 Работа с файлами и каталогами		2/1И	2		4	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.3, ПК-1.4, ПК-FS.3,ПК-FS.4, ПК-FS.5

2.5 Многопоточность и параллельное программирование		2/1И	6		6	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.3, ПК-1.4, ПК-FS.3,ПК-FS.4, ПК-FS.5
2.6 Основы LINQ	3	2/1И	4		4	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.3, ПК-FS.3, ПК-FS.4, ПК-FS.5, ПК-1.4
2.7 Работа с XML и JSON-форматами		2/1И	4		4	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.3, ПК-FS.4
2.8 Работа с датами, временем. Таймеры		2	2/0,2И		4,1	Изучение основной и дополнительной литературы по дисциплине, выполнение лабораторных работ	Тестирование, отчёт о выполнении лабораторной работы	ПК-1.2, ПК-1.3, ПК-1.4,ПК-FS.2, ПК-FS.3, ПК-FS.4, ПК-FS.5
Итого по разделу		16/8И	32/0,2И		34,1			
Итого за семестр		36/8И	54/16,2И		52,1		зачёт	
Итого по дисциплине		36/8И	54/16,2И		52,1		зачет	

5 Образовательные технологии

В ходе проведения занятий используются традиционные формы проведения занятий такие как:

1) информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами(монолог преподавателя).

2) лабораторная работа – организация учебной работы с реальными материальными и информационными объектами.

На лекционных и лабораторных работах используются технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов:

1) проблемная лекция – изложение материала, предполагающее постановку проблемных и дискуссионных вопросов, освещение различных научных подходов, авторские комментарии, связанные с различными моделями интерпретации изучаемого материала.

2) лабораторная работа на основе кейс-метода – обучение в контексте моделируемой ситуации, воспроизводящей реальные условия производственной,общественной деятельности. Обучающиеся должны проанализировать ситуацию,разобраться в сути проблем, предложить возможные решения и выбрать лучшее из них. Кейсы базируются на реальном фактическом материале или же приближены к реальной ситуации.

Применяются интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всехучастников, достижение на этой основе личностно значимого для нихобразовательного результата. Интерактивность подразумевает субъект-субъектные отношения в ходе образовательного процесса и, как следствие, формирование саморазвивающейся информационно-ресурсной среды.

Формы учебных занятий, используемые в дисциплине, с использованиемспециализированных интерактивных технологий:

1. Лекция «обратной связи» – лекция–провокация (изложение материала сзаранее запланированными ошибками), лекция-беседа, лекция-дискуссия, лекция пресс-конференция.

2. Лабораторная работа-дискуссия – коллективное обсуждение какого-либо спорного вопроса, проблемы, выявление мнений в группе (межгрупповой диалог,дискуссия как спор-диалог).

На лабораторных работах так же используются технологии проектного обучения, под которыми понимается организация образовательного процесса в соответствии с алгоритмом поэтапного решения проблемной задачи или выполнения учебного задания. Проект предполагает совместную учебно-познавательную деятельность группы студентов, направленную на выработку концепции, установление целей и задач, формулировку ожидаемых результатов, определение принципов и методик решения поставленных задач, планирование хода работы, поиск доступных и оптимальных ресурсов, поэтапную реализацию плана работы, презентацию результатов работы, их осмысление и рефлексию.

При обучении используются информационно-коммуникационные образовательные технологии, под которыми понимается организация образовательного процесса, основанная на применении специализированных программных сред и технических средств работы с информацией.

На всех лекциях изложение содержания сопровождается компьютернымипрезентациями, содержащими текстовые, иллюстративные, графические и видеоматериалы.

На лабораторных работах и во время самостоятельной работы обучающиесяработают с ресурсами и сервисами образовательного портала https://newlms.magtu.ru

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение

а) Основная литература:

1. Подбельский, В. В. Программирование. Базовый курс C# : учебник длявузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2025. — 369 с. —(Высшее образование). — ISBN 978-5-534-10616-9. — Текст : электронный //Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/560848

2. Галиаскаров, Э. Г. Анализ и проектирование систем с использованием UML :учебник для вузов / Э. Г. Галиаскаров, А. С. Воробьев. — Москва : ИздательствоЮрайт, 2025. — 125 с. — (Высшее образование). — ISBN 978-5-534-14903-6. —Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL:https://urait.ru/bcode/568178

3. Тузовский, А. Ф. Объектно-ориентированное программирование : учебникдля вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 213 с. —(Высшее образование). — ISBN 978-5-534-16316-2. — Текст : электронный //Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/561394

б) Дополнительная литература:

1. Тюкачев, Н. А. C#. Основы программирования / Н. А. Тюкачев, В. Г.Хлебостроев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 272 с. — ISBN 978-5-507-45438-9. — Текст : электронный // Лань : электронно-библиотечная система. —URL: https://e.lanbook.com/book/269840.

2. Цехановский, В. В. Управление данными : учебник / В. В. Цехановский, В. Д.Чертовской. — Санкт-Петербург : Лань, 2022. — 432 с. — ISBN 978-5-8114-1853-4. —Текст : электронный // Лань : электронно-библиотечная система. — URL:https://e.lanbook.com/book/212084.

в) Методические указания:

1. Варфоломеева, Т. Н. Практикум по основам алгоритмизаци и программирования [Электронный ресурс] : лабораторный практикум / Т. Н.Варфоломеева, С. А. Повитухин ; МГТУ. - Магнитогорск : МГТУ, 2016. - 1 электрон.опт. диск (CD-ROM). - Режим доступа:https://magtu.informsystema.ru/uploader/fileUpload?name=2407.pdf&show=dcatalogues/1/1130105/2407.pdf&view=true. - Макрообъект.

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

НаименованиеПО	№ договора	Срок действия лицензии
7Zip	свободно распространяемое ПО	бессрочно

MS Visual Studio2017 CommunityEdition	свободно распространяемое ПО	бессрочно
Браузер MozillaFirefox	свободно распространяемое ПО	бессрочно
Браузер Yandex	свободно	бессрочно

Профессиональные базы данных и информационные справочные

Название курса	Ссылка
Электронная база периодических изданий East ViewInformation Services, ООО «ИВИС»	https://dlib.eastview.com/
Национальная информационно-аналитическая система – Российский индекс научного	URL:https://elibrary.ru/project_risc.asp
Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»	URL: http://www1.fips.ru/
Российская Государственная библиотека. Каталоги	https://www.rsl.ru/ru/4readers/catalogues/
Электронные ресурсы библиотеки МГТУ им. Г.И.Носова	https://host.megaprolib.net/MP0109/Web

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Учебные аудитории для проведения занятий лекционного типаСпециализированная (учебная) мебель (столы, стулья, доска аудиторная),мультимедийное оборудование (проектор, компьютер, экран) для презентации учебного материала по дисциплине;

Учебные аудитории для проведения лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации Специализированная (учебная) мебель (столы, стулья, доска аудиторная),персональные компьютеры, объединенные в локальные сети с выходом в Internet и с доступом в электронную информационно-образовательную среду университета, оснащенные современными программно-методическими комплексами

Аудитории для самостоятельной работы (компьютерные классы; читальные залы библиотеки) Специализированная (учебная) мебель (столы, стулья, доска аудиторная), персональные компьютеры объединенные в локальные сети с выходом вInternet Internet и с доступом в электронную информационно-образовательную среду университета, оснащенные современными программно-методическими комплексами.

Помещение для хранения и профилактического обслуживания учебного оборудования Мебель (столы, стулья, стеллажи для хранения учебно-наглядных пособий и учебно-методической документации), персональные компьютеры.

Приложение 1

Учебно-методическое обеспечение самостоятельной работы обучающихся

Аудиторная самостоятельная работа студентов предполагает решение контрольных задач на лабораторных занятиях.

Раздел 1.

1. По известному радиусу определить длину окружности, площадь круга, площадь поверхности сферы и объем шара.
2. Дано натуральное K — количество секунд. Определить сколько это составляет часов, минут и секунд. Например, 4000 секунд — это 1 час, 6 минут и 40 секунд. Использовать не более 4-х арифметических операций.
3. Определить сколько лет понадобится шаху, чтобы собрать урожай зерна, требуемый изобретателем шахмат. Считать, что среднегодовой сбор составляет 70 млн. тонн, а на 1 грамм приходится 10 зерен.
4. Дан одномерный массив из N действительных случайных чисел в диапазоне от 1 до 50. Найти минимальный элемент среди элементов с нечетным индексом и максимальный среди элементов с четным.
5. Дан одномерный массив из N случайных действительных чисел в диапазоне от -4 до 8. Вывести в порядке невозрастания (убывания) элементы, модуль которых больше 2.
6. Для группы учащихся известны годовые оценки по следующим предметам: математика, физика, химия, информатика. Отобрать кандидатов на олимпиады (с отличными оценками) по каждому из предметов.
7. Для группы учащихся известны годовые оценки по следующим предметам: математика, физика, химия, информатика. Найти среднюю в группе оценку по каждому из предметов.
8. Известна ежемесячная заработная плата персонала предприятия в течение календарного года. Вывести фамилии сотрудников с минимальной и максимальной годовой заработной платой. Считать, что штат предприятия составляет 8 человек. Подсчет годовой зарплаты работника оформить в виде функции.
9. Дан одномерный массив из 100 случайных целых чисел в диапазоне от 5 до 25 включительно. Вывести все числа, которые максимально часто встречаются в массиве и количество их повторений. Подсчет количества повторений для числа оформить в виде функции.
10. Дан одномерный массив из 150 случайных целых чисел в диапазоне от 14 до 37 включительно. Вывести те числа, которые наиболее редко встречаются в массиве и количество их повторений. Подсчет количества повторений для числа оформить в виде функции.

Раздел 2.

1. Создать класс для работы со строками. Разработать следующие члены класса: – поле: StringBuilder line; – конструктор, позволяющий создать строку на основе заданного строкового литерала, и конструктор, позволяющий создавать пустую строку; – методы, позволяющие подсчитать количество пробелов в строке, 8 заменить в строке все прописные символы на строчные, удалить из строки все знаки препинания; – свойство, возвращающее общее количество элементов в строке (доступное только для чтения), и свойство, позволяющее установить значение поля в соответствии с введенным значением строки с клавиатуры, а также получить значение данного поля (доступно для чтения и записи); – индексатор, позволяющий по индексу обращаться к соответствующему символу строки; – перегрузку: операции унарного + (-) – преобразующей строку к строчным (прописным) символам; констант true и false – обращение к экземпляру класса дает значение true, если строка не пустая, иначе – false; операции & – возвращает значение true, если строковые поля двух объектов посимвольно равны (без учета регистра), иначе – false; операции

преобразования строки в тип StringBuilder (и наоборот).

2. Самостоятельно изучите тип данных DateTime, на основе которого необходимо создать класс для работы с датой. Данный класс должен содержать следующие члены класса: – поле DataTime data; – конструкторы, позволяющие установить заданную дату, дату 1.01.2000; – методы, позволяющие вычислить дату предыдущего дня, вычислить дату следующего дня, определить сколько дней осталось до конца месяца; – свойства, позволяющие установить или получить значение поле класса (доступно для чтения и записи), определить, является ли год високосным (доступно только для чтения); – индексатор, позволяющий определить дату i-го по счету дня относительно установленной даты (при отрицательных значениях индекса отсчет ведется в обратном порядке); – перегрузку: операции ! – возвращает значение true, если установленная дата не является последним днем месяца, иначе – false; констант true и false – обращение к экземпляру класса дает значение true, если установленная дата является началом года, иначе – false; операции & – возвращает значение true, если поля двух объектов равны, иначе false

3. Разработать программу с использованием делегатов. В программе следует: 1) определить делегат, принимающий несколько параметров различных типов и возвращающий значение некоторого типа; 2) написать не менее двух методов, соответствующих данному делегату; 3) написать метод, принимающий разработанный делегат, в качестве одного из входных параметров. Осуществить вызов метода, передавая ему в качестве параметра-делегата: – один из методов, разработанных в п. 2; – лямбда-выражение; 4) повторить п. 3, используя вместо разработанного делегата обобщенный делегат Func< > или Action< >, соответствующий сигнатуре разработанного делегата. 5) используя многоадресный делегат организовать цепочку вызовов методов, разработанных в п. 2; 6) создать блочное лямбда-выражение, соответствующее делегату, описанному в п. 1, и продемонстрировать его применение.

Раздел 3

Разработать консольное приложение согласно варианту. Запросы для поиска данных составлять с использованием технологии доступа к данным LINQ. 26 Набор данных реализовать с помощью динамических структур данных, используя обобщенный класс List – список (варианты заданий с чётными номерами) или обобщенный класс Queue – очередь (варианты заданий с чётными номерами). Запись в задании реализовать в виде класса.

1 Счет в банке представляет собой структуру с полями: номер счета, код счета, фамилия владельца, сумма на счете, дата открытия счета, годовой процент начисления. Поиск по номеру счета, дате открытия и владельцу.

2 Запись о товаре на складе представляет собой структуру с полями: номер склада, код товара, наименование товара, дата поступления на склад, срок хранения в днях, количество единиц товара, цена за единицу товара. Поиск по номеру склада, коду товара, дате поступления и сроку хранения (просроченные и непросроченные товары).

3 Запись о преподаваемой дисциплине представляется следующей структурой: код дисциплины в учебном плане, наименование дисциплины, фамилия преподавателя, код группы, количество студентов в группе, количество часов лекций, количество часов практики, наличие курсовой работы, вид итогового контроля (зачет или экзамен). Зачет – 0,35 ч на одного студента, экзамен – 0,5 ч на студента. Поиск осуществлять по фамилии преподавателя, коду группы, наличию курсовой, виду итогового контроля.

Приложение 2

Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ПК-1 Способен разрабатывать компьютерные игры, AR/VR -приложения -		
ПК-1.1	Проводит обследование организаций, выявляет информационные потребности пользователей, анализирует и формирует требования к мультимедийным приложениям	**Теоретические вопросы (к экзамену, зачету):** 1. Требования к алгоритмическому обеспечению приложений: виды, правила выделения 2. Требования к программному обеспечению приложений: виды, правила выделения 3. Требования к реализации пользовательского интерфейса приложения: виды, правила выделения **Практическое задание:** Выполнить анализ требований к компьютерной игре(по вариантам), которая будет разработана в качестве отчетного задания. Особенное внимание обратить требования к уровням сложности и реализации пользовательского интерфейса. **Комплексное задание:** Разработать концепцию компьютерной игры (по вариантам) в нотации UML (use case diagram)
ПК-1.2	Проектирует мультимедийные приложения (компьютерные игры и приложения виртуальной/дополненной реальности)	**Теоретические вопросы (к экзамену, зачету):** 1. Основы UML. 2. Диаграммы для представления концепции приложения 3. Структурные диаграммы UML 4. Поведенческие диаграммы UML **Практические задания:** Для заданной предметной области разработать диаграммы UML 1. Диаграмма классов. 2. Диаграмма объектов. 3. Диаграмма компонентов. 4. Диаграмма составной структуры. 5. Диаграмма вариантов использования. 6. Диаграмма последовательности. 7. Диаграмма коммуникации. 8. Диаграмма состояний. 9. Диаграмма деятельности. 10. Диаграмма размещения. 11. Диаграмма пакетов. 12. Временная диаграмма. 13. Диаграмма обзора взаимодействий. **Комплексное задание:** Разработать проект компьютерной игры (по вариантам) в нотации UML
ПК-1.3	Участвует в реализации проектов по	**Теоретические вопросы (к экзамену, зачету):** Нет **Практические задания:**

	созданию мультимедийных приложений под различные платформы и устройства	Нет **Комплексное задание:** По вариантам разработать одну из предложенных казуальных игр. Вариант 1. Игра "Сапёр". Вариант 2. Игра "Парные картинки". Вариант 3. Одиночная игра "Морской бой". Вариант 4. Игра "Пятнашки". Вариант 5. Игра "Жизнь". В игровом поле игрок случайным образом расставляет символы "*"- представителей популяции микроорганизмов. Затем по кнопке отслеживает изменение численности популяции по правилам: в пустой (мёртвой) клетке, с которой соседствуют три живые клетки, зарождается жизнь; если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае (если живых соседей меньше двух или больше трёх) клетка умирает («от одиночества» или «от перенаселённости»). Вариант 6. Игра "Нонограмма". Сбоку напротив каждой строки или столбца указывается количество закрашенных клеток (две цифры означают, что есть несколько участков указанной длины. Игрок должен правильно расставить точки, после этого программа должна вывести сообщение о победе. Ошибки в данной версии игры подсвечивать не надо! Квадратики можно заменить на символы "*". Вариант 7. Игра "Шашки" Вариант 8. Игра "Шахматы" Вариант 9. Игра "Тетрис" Вариант 10. Игра "2024" Вариант 11. Игра "Змейка" Вариант 12. Игра "Блоки"(аналог игры "Тетрис", только можно выбрать, куда поставить 1 из трёх выпавших блоков. Заполненный вертикальный или горизонтальный ряд исчезает) Общие требования: 1) графический интерфейс; 2) авторизация и статистика пользователей. Данные хранятся или в отдельном файле, или базе данных; 3) возможность выбора уровня сложности(3 режима): - для "Сапёра" - количество мин; размеры поля; - для "Парных картинок" - размер игрового поля; - для "Морского боя" - количество кораблей; - для "Пятнашек" - время, за которое нужно пройти игру; - для "Нонограммы" - размер поля; - для "Жизни" - количество видов популяции, тип ресурса(например, два вида и 1 ресурс (овцы, коровы, трава) или два неконкурирующих вида и 2 ресурса(зайцы, птицы, трава, ягоды) или два вида, 2 ресурса, при этом один из видов является ресурсом для второго (волки, зайцы, трава). По возможности использовать все возможности C#, изученные ранее.
ПК-1.4	Осуществляет тестирование мультимедийных приложений	**Теоретические вопросы (к зачету):** 1. Виды тестирования мультимедийных приложений 2. Unit-тестирование 3. Разработка Unit-тестов в среде Visual Studio: струкрура проекта, пересылка метода, параметры, определение результатов оценивания. **Практическое задание** 1. Разработать Unit-тесты в среде Visual Studio для математической программы из лабораторной работы 2 **Комплексное задание:**

		Разработать тесты для разрабатываемой компьютерной игры
ПК-FS – Разработка виртуальной и дополненной реальности		
ПК-FS.1	Знает основные понятия и концепции в области цифровых реальностей, основные сенсомоторные и психологические характеристики человека, включаемые в системы цифровых реальностей, виды и классификации систем цифровых реальностей	**Теоретические вопросы (к зачету):** 1. Психологические принципы разработки оконных пользовательских интерфейсов 2. Настройка визуальных характеристик элементов управления пользовательского интерфейса 3. Разработка индивидуального дизайна пользовательского интерфейса средствами платформы .Net **Практическое задание** 1. Создание непрямоугольной формы Windows Forms с помощью библиотеки GDI+ 2. Разработка анимационных элементов с помощью библиотеки GDI+. Примерное задание: «Разработайте приложение Windows Forms, которое будет отображать часы на клиентской области. Часы должны иметь стрелки: часы, минуты, секунды.» **Комплексное задание:** 1. Разработать грамотный с психологической точки зрения пользовательский интерфейс для итоговой игры (по вариантам). Учесть требования к цветовому оформлению, расположению компонентов формы и времени отклика приложения.
ПК-FS.2	Знает современные программные и аппаратные средства их реализации, проектные и технические процессы их создания, основные стандарты и методы оценивания пригодности при разработке систем цифровых реальностей, области и примеры их использования	**Теоретические вопросы (к зачету):** 1. Среды визуальной разработки. 2. MS VisualStudio 3. Project Rider 4. Eclipse, 5. Visual Studio Code 6. SharpDevelop IDE. **Практическое задание:** 1. Выполнить сравнительный анализ сред разработки объектно-ориентированных приложений на языке C# **Комплексное задание:** Выполнить обоснование выбора среды разработки компьютерной игры
ПК-FS.3	Умеет разрабатывать требования и архитектуру приложений на базе систем цифровых реальностей, выбирать технологии и	**Теоретические вопросы (к зачету):** 1. Среды визуальной разработки. 2. MS VisualStudio 3. Project Rider 4. Eclipse, 5. Visual Studio Code 6. SharpDevelop IDE. **Практическое задание:** 1. Выполнить сравнительный анализ сред разработки

	инструменты их реализации	объектно-ориентированных приложений на языке C# **Комплексное задание:** Выполнить обоснование выбора среды разработки компьютерной игры
ПК-FS.4	Умеет разрабатывать методы, модели, алгоритмы и программы приложений на базе систем цифровых реальностей, оценивать пригодность их использования, реализовывать проектные и технические процессы их создания	**Теоретические вопросы (к зачету):** 1. Элементы языка C#. Константы, идентификаторы, ключевые слова. 2. Типы данных и их объявление. 3. Выражения. Операнды и операции (унарные, бинарные, тернарные). Правила преобразования типов. 4. Операторы языка C#. Оператор выражение, составной оператор, операторы условного перехода. 5. Организация циклических вычислительных процессов с помощью операторов for, while, do while. 6. Организация ввода-вывода в языке C#. 7. Массивы в C# 8. Строки в C# 9. Классы в C# 10. Интерфейсы в C# 11. Делегаты в C# 12. Наследование в C# 13. Реализация полиморфизма в C# 14. Работа с наборами данных: массивы и коллекции 15. Потоки ввода- вывода в C# 16. Создание таймеров 17. Задачи в C# 18. Язык запросов LINQ 19. Работа с XML-файлами в C# 20. Работа с датами и временем в C# **Практические задания** 1. Создать класс Point3D, содержащий следующие члены класса: – поля: int x, y, z; – конструкторы, позволяющие создать экземпляр класса с нулевыми координатами, с заданными координатами; – методы, позволяющие вывести координаты точки на экран, рассчитать расстояние от начала координат до точки, переместить точку на плоскости на вектор (a, b,c); – свойства, позволяющие получить/установить координаты точки (доступное для чтения и записи), умножить координаты точки на скаляр (доступное только для записи); – индексатор, позволяющий по индексу 0 обращаться к полю x, по индексу 1 –к полю y, 2- к полю z; при других значениях индекса выдается сообщение об ошибке; – перегрузку: - операции ++ (- -) – одновременно увеличивает (уменьшает) значение полей на 1; --констант true и false – обращение к экземпляру класса дает значение true, если значение полей x и y совпадает, иначе false; - операции бинарный + – одновременно добавляет к полям значение скаляра. 2. Создать абстрактный класс Pair (пара значений) с виртуальными арифметическими операциями и методом вывода на экран. На его основе реализовать классы Money (деньги) и Complex (комплексное число). В классе Money денежная сумма представляется в виде двух целых, в которых хранятся рубли и копейки соответственно. При выводе части числа снабжаются

словами «руб.» и «коп.». В классе Complex предусмотреть при выводе символ мнимой части (i). Создать класс Series (набор), содержащий список (или массив) объектов этих классов в динамической памяти. Предусмотреть возможность вывода объектов списка. Написать демонстрационную программу, в которой будут использоваться все методы классов.

3. На основе данных входного файла составить список студентов группы, включив следующие данные: ФИО, год рождения, домашний адрес, какую школу окончил. Вывести в новый файл информацию о студентах, окончивших заданную школу, отсортировав их по году рождения. Подразумевается, что исходная информация хранится в текстовом файле input.txt, каждая строка которого содержит полную информацию о некотором объекте, результирующая информация должна быть записана в файл output.txt. Для хранения данных внутри программы организовать массив структур. В типе структура реализуется метод CompareTo интерфейса IComparable, перегружается метод ToString базового класса object и необходимые операции отношения, поля данных и дополнительные методы продумайте самостоятельно.

4. Реализуйте пользовательский тип делегата требуемой сигнатуры и выполните с его использованием вызов нескольких методов (с корректной сигнатурой): Action<Func<float>, bool, List<float>>

5. Создайте приложение на языке C# в MS Visual Studio. В соответствии с индивидуальным вариантом разработайте требуемый тип делегата (пользовательский, библиотечный или лямбда- выражение). Реализуйте асинхронное выполнение метода на основе разработанного делегата с возможностью мониторинга процесса выполнения, передачи параметров в метод и получения результата работы метода. Измените приложение так, чтобы использовался тайм-аут. Реализуйте механизм вывода информации в консоль о ходе решения задачи асинхронным методом.

Вариант	**Тип делегата**	**Решаемая задача (результат метода)**	**Входные параметры**
1	пользовательский	Метод возвращает сумму элементов матрицы целых случайных чисел	Два параметра: размер матрицы
2	библиотечный	Метод возвращает разницу максимального и минимального элементов матрицы целых случайных чисел	Два параметра: размер матрицы
3	лямбда-выражение	Метод возвращает логическоезначение, указывающее существует ли заданное число в массиве целыхслучайных чисел	Два параметра: размер массива и искомый элемент

6. Создайте консольное приложение с использованием MS Visual Studio. Реализуйте метод для запуска в отдельном потоке (в соответствии с индивидуальным вариантом). Создайте делегат для представления метода (если требуется).В основной

программе (функция Main( )) реализуйте создание массива потоков (размер определите самостоятельно). Затем запустите все элементы массива (потоки) на выполнение.Метод, выполняющийся в параллельных потоках, должен выводить информацию о ходе своего выполнения в консоль приложения.

Вариант	Метод для реализации
1	Метод вычисления среднего арифметического элементов массива.
2	Метод поиска максимального элемента в матрице.
3	Метод преобразования матрицы случайных чисел: каждый элемент заменяется косинусом элемента матрицы.

7. Разработайте классы для решения задачи в соответствии с индивидуальным вариантом с использованием пула потоков. В каждом задании необходимо разработать коллекцию элементов заданного типа и класс для управления коллекцией, осуществляющий обработку элемента с использованием пула потоков. В алгоритме решения задачи предусмотрите задержку алгоритма с использованием метода Thread.Sleep( ). Сделайте вывод: как меняется поведение программы при изменении времени искусственной задержки алгоритма.

Вариант	Задача, решаемая с использованием пула потоков (метод обработки элемента)	Тип элемента коллекции
1.	Метод находит среднее арифметическое элементов матрицы случайных чисел.	Класс определяет матрицу (размер выбирается случайным образом)
2.	Метод находит результат шифрования строки: каждый исходный символ строки заменяется шифрованным символом, код которого на n больше кода исходного символа.	Класс представляет пару строк: исходная и шифрованная (исходная задается в конструкторе)
3.	Метод находит подмножество элементов массиваслучайных чисел, которые делятся на 6	Класс представляетмассив случайных чисел случайного размера

8. Создайте консольное приложение в среде Visual Studio. Реализуйте решение задачи в соответствии с применением класса Task. Реализуйте решение задачи с использованием механизма задач продолжения в соответствии с вариантом индивидуального задания

Варианты

1 Генерация матрицы случайных чисел (размер задается пользователем); 2 задачи: расчет суммы элементов; поиск максимального элемента.

2 Генерация массива случайных чисел (размер задается пользователем);2 задачи: вычисление количества элементов, делящихся на 3; поиск минимального элемента.

3 Генерация матрицы случайных чисел (размер определяется

		случайным образом);2 задачи: расчет суммы всехчетных элементов; поиск среднего арифметическогоэлементов. 9. Из массива целых чисел X[] требуется выбрать числа, значения которых >= 4, и записать эти числа в список Y, отсортировав выбранные числа по возрастанию. 10. Даны сведения о 10 сотрудниках: ФИО, должность, оклад. Выбрать сотрудников с окладом больше 10000 11. Имеются сведения о студентах: ФИО, группа, размер стипендии. Вывести средний размер стипендии по каждой группе. 12. Создать XML файл, содержащий сведения о 5 студентах и их оценках по 5 предметам. Используя созданный XML-файл, выполнить следующие задания: выбрать и вывести информацию о каком-либо студенте;выбрать студентов со средним баллом меньше 3. **Комплексные задания:** Вариант 1. Песочные часы. Программа должна отображать песочные часы. При нажатии кнопки "Старт" часы должны перевернуться и из верхней части должен начать пересыпаться песок в нижнюю часть. Время пересыпания песка должно задаваться с помощью ComboBox-а: 1, 3, 5 минут. Скорость пересыпания песка зависит от времени работы часов. Под часами должна находиться строка, в которой отображается количество оставшегося времени работы часов. Предусмотреть кнопку "Сброс", останавливающую таймер и возвращающую часы в исходное нерабочее состояние(весь песок внизу). Вариант 2. Клепсидра. Программа должна отображать водяные часы (клепсидру). При нажатии кнопки "Старт" вода из верхней части часов должна капать вниз. Время работы часов должно задаваться с помощью ComboBox-а: 1, 3, 5 минут. Под часами должна находиться строка, в которой отображается время работы часов. Предусмотреть кнопку "Сброс", останавливающую таймер и возвращающую часы в исходное нерабочее состояние. Предусмотреть изменение скорости капель в зависимость от времени работы часов. Вариант 3. Сугроб. При запуске программы на рабочем поле сверху вниз падают снежинки, каждый раз разное количество(задаётся случайным образом). Когда снежинка касается края снежного покрова, она разрушается, рассыпаясь в снежную пыль. Когда рабочее поле заполнится снегом на 20% от его высоты, появляется снеговик. При появлении 5 снеговиков поле обнуляется. Скорость появления снежинок регулируется с помощью ползунка внизу формы. Состояние рабочего поля запоминается и восстанавливается при каждом запуске программы. Вариант 4. Монетки. По кнопке "Старт" запускается программа визуализации накопления капитала: в строке в правом верхнем углу отображается накапливаемая сумма, слева начинают падать монетки и собираться в гору. Когда высота горы достигнет 50% от высоты формы, изображение должно уменьшиться с сохранением масштаба. Скорость накопления можно менять. История работы сохраняется, и при следующем запуске программы счётчик должен показывать накопленную сумму с учётом времени, которое прошло с момента нажатия кнопки старт. Предусмотреть возможность сброса

		накопления - в этом случае счётчик обнуляется и не увеличивается до нажатия кнопки "Старт". Вариант 5. Белка. На рабочем поле отображается колесо, в котором бежит белка. Каждый оборот колеса добавляет 0,5 Вт к счетчику энергии, который отображается над колесом. Белка бегает только во время работы программы, состояние счётчика при этом сохраняется. Каждый 1кВт белка получается награду - 1 орех. Награда отображается сбоку от колеса. При завершении работы программы все орехи съедаются. Вариант 6. Конвейер. На рабочем поле изображен конвейер, с которого в корзинку падают монетки. Количество монет разное в отдельный момент времени. Каждые 200 монет корзинка опустошается. Над конвейером располагается строка, в которой отображается количество собранных монеток. Скорость работы конвейера регулируется ползунком, который расположен снизу. Конвейер не прекращает работать даже при выходе из программы, т.е. при новом запуске должна отобразиться сумма с начала запуска конвейера. Дату и время запуска выбрать произвольные. Вариант 7. Беговая дорожка. На рабочем поле изображена беговая дорожка и тренирующийся спортсмен. Над дорожкой - счётчик расхода калорий. Параметры спортсмена (рос, вес) редактируются справа. Угол наклона и скорость дорожки можно менять - скорость спортсмена меняется тоже. Расход калорий должен меняться в зависимости от параметров спортсмена и состояния дорожки. При завершении результаты тренировки записываются в файл, а счётчик обнуляется. Вариант 8. Листопад. При запуске программы на рабочем поле сверху вниз падают желтые листья, каждый раз разное количество(задаётся случайным образом). На земле все листочки лежат на широкой стороне. Когда рабочее поле заполнится листьями на 10% от его высоты, появляется куча листьев. При появлении 3 куч поле обнуляется. Скорость появления листьев регулируется с помощью ползунка внизу формы. Состояние рабочего поля запоминается и восстанавливается при каждом запуске программы. Вариант 9 Утечка. Из старой трубы в ведро капает вода. Скорость наполнения ведра регулируется с помощью ползунка слева от ведра. При заполнении вода выливается и процесс начинается заново. Над ведром располагаются 2 строки: количество воды в кубометрах(дробь) и стоимость утекающей воды(в рублях). Вода капает постоянно, вне зависимости от работы программы. При новом запуске счетчики должны отражать расход воды и денег с учётом всего прошедшего времени. Вариант 10 Просмотры . Программа имитирует рост просмотров некоторого ресурса. Количество просмотров каждую секунду разное. Суммарное количество просмотров располагается в правом верхнем углу. Под количеством - график, обновляющийся каждую 1, 2 или 5 секунд (выбрать с помощью ComboBox). Статистка просмотров сохраняется и при новом запуске программы рост просмотров возобновляется. Вариант 11. Я-миллионер. По кнопке "Старт" запускается программа визуализации накопления капитала: в строке в правом верхнем углу отображается накапливаемая сумма, слева начинают денежные купюры и собираться в гору. Когда высота горы достигнет 50% от высоты формы, изображение должно уменьшиться с сохранением масштаба. Скорость

		накопления можно менять. История работы сохраняется, и при следующем запуске программы счётчик должен показывать накопленную сумму с учётом времени, которое прошло с момента нажатия кнопки старт. Предусмотреть возможность сброса накопления - в этом случае счётчик обнуляется и не увеличивается до нажатия кнопки "Старт". Вариант 12. Конфеты. На рабочем поле изображен конвейер, с которого в корзинку падают конфеты. Количество конфет разное в отдельный момент времени. Каждые 100 конфет корзинка опустошается. Над конвейером располагается строка, в которой отображается количество собранных конфет. Скорость работы конвейера регулируется ползунком, который расположен снизу. Конвейер не прекращает работать даже при выходе из программы, т.е. при новом запуске должна отобразиться сумма с начала запуска конвейера. Дату и время запуска выбрать произвольные.
ПК-FS.5	Владеет навыками разработки и оценки приложений на базе систем цифровых реальностей с использованием современных аппаратных и программных средств	**Практические задания** 1. Разработка простого приложения с графическим интерфейсом. Программа должна вывести надпись с просьбой представиться, предусмотреть поле для ввода имени, кнопку «ОК». При нажатии на кнопку программа выводит надпись: «Привет, «Имя»!» 2. Реализовать приложение с графическим пользовательским интерфейсом, которое решает следующую задачу (по вариантам). Вариант 1. Написать программу, которая по названию геометрической фигуры (треугольник, квадрат, трапеция и т.п.) позволяет ввести нужное количество параметров для вычисления периметра, вычислить периметр и вывести результат в объект типа Label. Вариант 2. Написать программу, которая по названию системы счисления (двоичная, восьмеричная, десятичная, шестнадцатеричная) формирует доступные символы для ввода числа, позволяет ввести и отображает во всех заданных системах счисления. 3. Разработать казуальную несложную игру. Состояние игрового поля хранится в двумерном массиве. Общие требования: 1) графический интерфейс; 2) авторизация и статистика пользователей. Данные хранятся или в отдельном файле, или базе данных; 3) возможность выбора уровня сложности(3 режима): по возможности использовать все возможности C#, изученные ранее. **Комплексное задание:** По вариантам разработать одну из предложенных казуальных игр. Вариант 1. Игра "Сапёр". Вариант 2. Игра "Парные картинки". Вариант 3. Одиночная игра "Морской бой". Вариант 4. Игра "Пятнашки". Вариант 5. Игра "Жизнь". В игровом поле игрок случайным образом расставляет символы "*"- представителей популяции микроорганизмов. Затем по кнопке отслеживает изменение численности популяции по правилам: в пустой (мёртвой) клетке, с которой соседствуют три живые клетки, зарождается жизнь; если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае (если живых соседей меньше двух или больше трёх) клетка умирает («от одиночества» или «от перенаселённости»). Вариант 6. Игра "Нонограмма". Сбоку напротив каждой строки или столбца указывается количество закрашенных клеток (две цифры

		означают, что есть несколько участков указанной длины. Игрок должен правильно расставить точки, после этого программа должна вывести сообщение о победе. Ошибки в данной версии игры подсвечивать не надо! Квадратики можно заменить на символы "*". Вариант 7. Игра "Шашки" Вариант 8. Игра "Шахматы" Вариант 9. Игра "Тетрис" Вариант 10. Игра "2024" Вариант 11. Игра "Змейка" Вариант 12. Игра "Блоки"(аналог игры "Тетрис", только можно выбрать, куда поставить 1 из трёх выпавших блоков. Заполненный вертикальный или горизонтальный ряд исчезает) Общие требования: 1) графический интерфейс; 2) авторизация и статистика пользователей. Данные хранятся или в отдельном файле, или базе данных; 3) возможность выбора уровня сложности(3 режима): - для "Сапёра" - количество мин; размеры поля; - для "Парных картинок" - размер игрового поля; - для "Морского боя" - количество кораблей; - для "Пятнашек" - время, за которое нужно пройти игру; - для "Нонограммы" - размер поля; - для "Жизни" - количество видов популяции, тип ресурса(например, два вида и 1 ресурс (овцы, коровы, трава) или два неконкурирующих вида и 2 ресурса(зайцы, птицы, трава, ягоды) или два вида, 2 ресурса, при этом один из видов является ресурсом для второго (волки, зайцы, трава). По возможности использовать все возможности C#, изученные ранее.
ПК-FS.6	Владеет навыками применения стандартов при составлении технической документации на разработку, испытание и использование приложений на базе систем цифровых реальностей	**Теоретические вопросы (к экзамену, зачету):** 1. История развития стандарта стандарта C# 2. Обзор спецификации проекта C# (по версии стандарта ECMA 8) 3. Обзор спецификации компонентов C# (по версии стандарта ECMA 8) 4. Направления развития стандарта C# 5. Корпоративные стандарты оформления программного кода: именования классов и его составляющих, комментариев. **Практическое задание:** Разработайте стандарт оформления программного кода (листинг - основной компонент программной документации), указав правила оформления программного кода: именования классов и его составляющих, комментариев. **Комплексные задания:** 1. Для итоговой игры (по вариантам) подготовить правила оформления программного кода с учётом стандарт C# ECMA8 2. Для итоговой игры (по вариантам) оформить программный ко в соответствии с разработанными правилами

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине проводится в форме зачёта в 3 семестре.

Требования к зачёту по дисциплине

Зачет выставляется на последнем занятии в семестре преподавателем, который ведёт лабораторные занятия при условии выполнения следующих требований.

1. Пройден курс «Основы программирования на C#»(https://stepik.org/course/64831/promo?search=1288415581) и выложено подтверждение
2. Выполнены и сданы все лабораторные работы в семестре.
3. Пройдены все тесты по лекциям за семестр не ниже, чем на 90%.
4. Успешная защита итогового проекта .

Показатели и критерии оценивания:

- на оценку «зачтено» – обучающийся демонстрирует достаточно высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности;
- на оценку «не зачтено» – обучающийся не демонстрирует того, что должен продемонстрировать на оценку «зачтено»