ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 4E0A9027EE0457E6D0F0E1B1383CC6F8
Владелец: Корабельников Сергей Кимович
Действителен: с 23.09.2026 до 17.12.2027

Правительство Санкт-Петербурга
Комитет по науке и высшей школе

Санкт-Петербургское государственное бюджетное
профессиональное образовательное учреждение
«АКАДЕМИЯ ТРАНСПОРТНЫХ ТЕХНОЛОГИЙ
имени Героя Социалистического Труда И.Г. Зубкова»

ПРИНЯТО
на заседании педагогического совета
Протокол
от «27» мая 2026 г.
№ 5

УТВЕРЖДЕНО
Приказом директора
СПб ГБПОУ «АТТ имени Героя Социалистического Труда И.Г. Зубкова»
от «__» _____ 202_ г.
№ _____

ПРОГРАММА ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Наименование: **Разработка приложений на языке C# с использованием .NET**

Вид: **программа повышения квалификации**

Категория слушателей: **лица, получающие и/или имеющие среднее профессиональное образование, высшее образование**

Срок реализации программы: **6 недель**

Объём программы: **26 часов**

Форма обучения: **заочная (с применением электронного обучения и дистанционных образовательных технологий)**

г. Санкт-Петербург
2026 г.

Разработано:

Преподавателем СПб ГБПОУ «АТТ имени Героя Социалистического Труда И.Г. Зубкова» Черновой А.А.

Рассмотрено и одобрено
на заседании цикловой комиссии № 5 «Информационные технологии» СПб ГБПОУ «АТТ имени Героя Социалистического Труда И.Г. Зубкова»
Протокол № 1 от 27 августа 2026 г.
Председатель ЦК Чернова А.А.

Проверено

Методист Алексеенкова П.А.

Зав. методическим кабинетом Мельникова Е.В.

Рекомендовано и одобрено:
Методическим советом СПб ГБПОУ «АТТ имени Героя Социалистического Труда И.Г. Зубкова»
Протокол №5 от 27 мая 2026 г.
Председатель Методического совета Вишневская М.В.,
заместитель директора по учебной работе

Содержание

1. Цель реализации программы	4
2. Объем программы	4
3. Срок реализации программы	4
4. Форма обучения	4
5. Форма реализации программы	4
6. Нормативно-правовая основа разработки программы	4
7. Категория обучающихся	4
8. Выдаваемый документ о квалификации	4
9. Планируемые результаты освоения программы	4
10. Учебный план	7
11. Календарный учебный график	8
12. Рабочая программа	9
13. Организационно-педагогические условия реализации рабочей программы	12
14. Контроль и оценка результатов освоения программы	12

Программа дополнительного профессионального образования
Разработка приложений на языке C# с использованием .NET

Актуальность программы дополнительного профессионального образования по разработке приложений на C# с использованием .NET заключается в сочетании факторов: востребованности технологии на рынке труда, её роли в импортозамещении, кроссплатформенности, активному развитию экосистемы, универсальности применения и интеграции с российскими IT-стандартами. Такая программа позволит специалистам повысить квалификацию и стать востребованными на рынке, а компаниям — получить кадры для решения ключевых задач цифровой трансформации.

1. Целью реализации программы является совершенствование профессиональных компетенций в области разработки приложений на языке C# с использованием платформы .NET, необходимых для выполнения трудовых функций программиста.

2. Объем программы: 26 часов

3. Срок реализации программы: 6 недель

4. Форма обучения: заочная (с применением электронного обучения и дистанционных образовательных технологий)

5. Форма реализации программы: теоретические и практические занятия с применением электронного обучения и дистанционных образовательных технологий

6. Нормативно-правовая основа разработки программы:

– Федеральный закон «Об образовании в Российской Федерации» от 29 декабря 2012 года №273-ФЗ с изменениями и дополнениями;

– Приказ Министерства науки и высшего образования Российской Федерации от 24 марта 2025 г. № 266 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;

– Профессиональный стандарта 06.001 «Программист», утвержденный Приказом Министерства труда и социальной защиты РФ от 20 июля 2022 г. №424.

7. Категория обучающихся

Лица, имеющие или получающие среднее профессиональное и (или) высшее образование. Медицинские ограничения регламентированы Перечнем медицинских противопоказаний Минздрава России.

8. Выдаваемый документ о квалификации:

Лицам, успешно освоившим соответствующую дополнительную профессиональную программу и прошедшим итоговую аттестацию, выдается документ: удостоверение о повышении квалификации «Разработка приложений на языке C# с использованием .NET».

9. Планируемые результаты освоения программы

В результате освоения программы слушатель должен приобрести следующие знания и умения, необходимые для формирования у слушателей компетенции, указанной в п. 1.2.

Вид деятельности	**Разработка приложений на языке C# с использованием .NET**		
Профессиональные компетенции/ Трудовые функции	Слушатель должен **знать:**	Слушатель должен **уметь:**	Слушатель должен **владеть навыками/ трудовыми действиями**
ПК 1.1 Написание программного кода с использованием языков программирования, определения и манипулирования	З1 - Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные	У1 - Применять выбранные языки программирования для написания программного кода У2 - Использовать выбранную среду программирования	ТД1 - Создание программного кода в соответствии с техническим заданием (готовыми спецификациями) ТД2 - Оптимизация программного кода с

Вид деятельности	**Разработка приложений на языке C# с использованием .NET**		
Профессиональные компетенции/ Трудовые функции	Слушатель должен **знать:**	Слушатель должен **уметь:**	Слушатель должен **владеть навыками/ трудовыми действиями**
данными в базах данных	библиотеки языка программирования З2 - Технологии программирования З3 - Компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними	и средства системы управления базами данных	использованием специализированных программных средств
ПК.1.2 Оформление программного кода в соответствии с установленными требованиями	З4 - Инструментарий для создания и актуализации исходных текстов программ Методы повышения читаемости программного кода Системы кодировки символов, форматы хранения исходных текстов программ	У3 - Применять инструментарий для создания и актуализации исходных текстов программ	ТД3 - Приведение наименований переменных, функций, классов, структур данных и файлов в соответствие с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода ТД4 - Структурирование исходного программного кода в соответствии с нормативно-техническими документами (стандартами и регламентами), определяющими требования к оформлению программного кода
ПМ.1.3 Проверка и отладка программного кода	З5 - Методы и приемы отладки программного кода З6 - Типы и форматы сообщений об	У4 - Выявлять ошибки в программном коде У5 - Применять методы и приемы	ТД5 - Анализ и проверка исходного программного кода

Вид деятельности	**Разработка приложений на языке C# с использованием .NET**		
Профессиональные компетенции/ Трудовые функции	Слушатель должен **знать:**	Слушатель должен **уметь:**	Слушатель должен **владеть навыками/ трудовыми действиями**
	ошибках, предупреждений З7 - Современные компиляторы, отладчики и оптимизаторы программного кода	отладки программного кода У6 - Применять современные компиляторы, отладчики и оптимизаторы программного кода	

Освоение программы направлено на достижение личностных результатов:

ЛР4 — осознающий ценность собственного труда, стремящийся к формированию конструктивного «цифрового следа»;

ЛР10 — заботящийся о собственной и чужой безопасности, в том числе цифровой;

ЛР13 — ответственный, дисциплинированный, нацеленный на достижение задач и эффективное взаимодействие в команде;

ЛР14 — владеющий навыками оценки достоверности информации в цифровой среде и построения логических умозаключений на основе данных;

ЛР25 — осуществляющий поиск, анализ и интерпретацию информации для выполнения задач профессиональной деятельности;

ЛР31 — активно применяющий полученные знания на практике;

ЛР32 — способный анализировать производственную ситуацию и быстро принимать решения;

ЛР34 — выбирающий способы решения задач профессиональной деятельности применительно к различным контекстам;

ЛР35 — использующий информационные технологии в профессиональной деятельности.

10. Учебный план

Наименование: «Разработка приложений на языке C# с использованием .NET»

Вид: дополнительная профессиональная программа повышения квалификации

Категория слушателей: лица, получающие и/или имеющие среднее профессиональное образование, высшее образование

Срок освоения программы: 6 недель

Объём программы: 26 часов

Форма обучения*:* заочная (с применением электронного обучения и дистанционных образовательных технологий)

№ п/п	Наименование раздела/модуля/ темы занятия	Всего (академический час)	Самостоятельная работа	Количество часов, отведенных на аттестацию	Аудиторные занятия		Занятия с использованием ЭО и ДОТ час.		Форма аттестации (формы контроля)
					из них		из них		
					теоретические обучение	Практическая подготовка*	теоретические обучение	Практическая подготовка*	
1	Тема 1. Основы языка C# и среда разработки	2	-	-	1	1	-	-	-
2	Тема 2. Объектно-ориентированное программирование в C#	4	-	-	-	-	1	3	-
3	Тема 3. Коллекции, LINQ и работа с исключениями	4	-	-	-	-	1	3	-
4	Тема 4. Работа с файлами и сериализация данных	4	-	-	-	-	1	3	-
5	Тема 5. Разработка оконных приложений (Windows Forms)	4	-	-	-	-	1	3	-
6	Тема 6. Асинхронное программирование и Entity Framework Core	4	-	-	-	-	1	3	-
7	**Консультация**	2	-	-	**2**	-	-	-	-
8	**Итоговая аттестация в форме зачета**	**2**	-	-	-	**2**	-	-	**зачет**
9	**ИТОГО по программе**	**26**							

11. Календарный учебный график

Наименование: Разработка приложений на языке C# с использованием .NET

Вид: дополнительная профессиональная программа повышения квалификации

Категория слушателей: лица, получающие и/или имеющие среднее профессиональное образование, высшее образование

Срок освоения программы: 6 недель

Объём программы: 26 часов

Форма обучения: заочная (с применением электронного обучения и дистанционных образовательных технологий)

Наименование тем	**Распределение учебной нагрузки по периодам обучения**					
	1	**2**	**3**	**4**	**5**	**6**
Тема 1. Основы языка C# и среда разработки	2	-	-	-	-	-
Тема 2. Объектно-ориентированное программирование в C#	2	2	-	-	-	-
Тема 3. Коллекции, LINQ и работа с исключениями	-	2	2	-	-	-
Тема 4. Работа с файлами и сериализация данных	-	-	2	2	-	-
Тема 5. Разработка оконных приложений (Windows Forms)	-	-	-	2	2	-
Тема 6. Асинхронное программирование и Entity Framework Core	-	-	-	-	2	2
Консультация	-	-	-	-	-	2
Итоговая аттестация в форме зачета	-	-	-	-	-	2
ИТОГО по программе	**4**	**4**	**4**	**4**	**4**	**6**

12. Рабочая программа

Наименование: Разработка приложений на языке C# с использованием .NET

Вид: дополнительная профессиональная программа повышения квалификации

Категория слушателей: лица, получающие и/или имеющие среднее профессиональное образование, высшее образование

Срок освоения программы: 6 недель

Объём программы: 26 часов

Форма обучения: заочная (с применением электронного обучения и дистанционных образовательных технологий)

№ занятия	Наименование разделов и тем. Содержание учебных занятий. Формы организации деятельности обучающихся	Объем часов	Литература, наглядные пособия, оборудование, ЭОР, программное обеспечение
1.	**Тема 1. Основы языка C# и среда разработки.** Синтаксис C#: типы данных, переменные, операторы, условные конструкции (if-else, switch), циклы (for, while, foreach). Обзор Visual Studio 2022	1	О1, О2, Visual Studio 2022, документация Microsoft .NET
	Практическое занятие №1 Разработка консольного приложения «Калькулятор». Создание консольного приложения. Ввод данных, реализация арифметических операций, обработка пользовательского ввода, форматированный вывод результатов	1	О1, Visual Studio 2022
2.	**Дистанционное занятие** **Тема 2. Объектно-ориентированное программирование в C#.** Классы, объекты, конструкторы, свойства. Принципы ООП: инкапсуляция, наследование, полиморфизм. Интерфейсы и абстрактные классы	1	О1, Д1, Visual Studio 2022
3.	**Дистанционное занятие** **Практическое занятие № Проектирование иерархии классов для системы «Библиотека».** Создание базового класса Book, производных классов PrintedBook и EBook. Реализация интерфейса ILendable с методами Borrow() и Return(). Демонстрация полиморфизма	3	О1, Visual Studio 2022
4.	**Дистанционное занятие** **Тема 3. Коллекции, LINQ и работа с исключениями.** Обзор коллекций: List, Dictionary, HashSet. Основы LINQ: фильтрация (Where), проекция (Select), сортировка, группировка, агрегация. Обработка исключений: try-catch-finally, пользовательские исключения	1	О1, О2, Visual Studio 2022

№ занятия	Наименование разделов и тем. Содержание учебных занятий. Формы организации деятельности обучающихся	Объем часов	Литература, наглядные пособия, оборудование, ЭОР, программное обеспечение
5.	**Дистанционное занятие** **Практическое занятие №3 Обработка данных с LINQ и управление исключениями.** Создание коллекции объектов Book. Реализация LINQ-запросов: фильтр по жанру, группировка по автору, вычисление средней цены. Добавление обработки исключений при некорректных данных	3	О1, О2, Visual Studio 2022
6.	**Дистанционное занятие** **Тема 4. Работа с файлами и сериализация данных.** Классы File, StreamReader, StreamWriter. Чтение и запись текстовых файлов. Сериализация и десериализация объектов в JSON с использованием System.Text.	1	О1, О2, Visual Studio 2022, документация System.Text.Json
7.	**Дистанционное занятие** **Практическое занятие №4 Сохранение и загрузка данных библиотеки.** Реализация методов сохранения коллекции книг в JSON-файл и загрузки из файла. Создание консольного меню: просмотр всех книг, добавление новой книги, сохранение в файл, загрузка из файла	3	О1, Д1, Visual Studio 2022
8.	**Дистанционное занятие** **Тема 5. Разработка оконных приложений (Windows Forms).** Основы Windows Forms: форма, элементы управления (Button, TextBox, Label, DataGridView, ComboBox). Обработка событий. Привязка данных к DataGridView	1	О2, MS SQL Server
9.	**Дистанционное занятие** **Практическое занятие №5 Создание оконного приложения «Каталог книг».** Разработка формы с таблицей книг (DataGridView), полями ввода (название, автор, жанр, цена) и кнопками «Добавить», «Удалить», «Сохранить в файл», «Загрузить из файла». Реализация обработчиков событий	3	О1, Visual Studio 2022
10.	**Дистанционное занятие** **Тема 6. Асинхронное программирование и Entity Framework Core.** Понятие асинхронности: async/await, класс Task. Основы Entity Framework Core: DbContext, DbSet, Code First. Миграции. CRUD-операции с БД	1	О1, О2, Visual Studio 2022, MS SQL Server, документация EF Core
11.	**Дистанционное занятие** **Практическое занятие №6. Подключение базы данных к приложению «Каталог книг».** Создание контекста БД (AppDbContext). Настройка строки подключения. Выполнение миграции. Замена файлового хранилища на базу данных. Реализация асинхронной загрузки данных в DataGridView	3	О1, Visual Studio 2022, MS SQL Server, EF Core

№ занятия	Наименование разделов и тем. Содержание учебных занятий. Формы организации деятельности обучающихся	Объем часов	Литература, наглядные пособия, оборудование, ЭОР, программное обеспечение
12.	**Консультация** Разбор типовых ошибок при разработке на C#. Повторение ключевых тем: ООП, LINQ, сериализация, Windows Forms, EF Core. Ответы на вопросы	2	О1, О2, Д1, интернет-ресурсы
13.	**Итоговая аттестация в форме зачета** Выполнение практического задания: разработка мини-проекта «Каталог товаров» (консольное и оконное приложение с сохранением в БД)	2	Visual Studio 2022, MS SQL Server
	ИТОГО по программе	**26**	

13. Организационно-педагогические условия реализации рабочей программы

13.1 Материально-техническое обеспечение программы

Реализация программы предполагает наличие материально-технического обеспечения:

– аудитория, оборудованная видеопроекционным оборудованием и экраном, имеющая выход в Интернет;

– лаборатория Программирования и баз данных;

– учебное программное обеспечение – Visual Studio 2022 Community, .NET SDK 8.0, MS SQL Server Express.

13.2 Информационные ресурсы

Для реализации программы библиотечный фонд образовательной организации имеет печатные и/или электронные образовательные и информационные ресурсы, рекомендуемые для использования в образовательном процессе.

Основная литература:

1) О1 Троелсен, Э. Язык программирования C# 10 и платформа .NET 6 / Э. Троелсен, Ф. Джепикс. — Москва: Диалектика, 2023. — 1488 с. — ISBN 978-5-907515-48-2.

2) О2 Албахари, Д. C# 10. Справочник. Полное описание языка / Д. Албахари, Б. Албахари. — Москва: Диалектика, 2023. — 1056 с. — ISBN 978-5-907515-44-4.

Дополнительная литература:

1) Д1 Рихтер, Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Д. Рихтер. — Санкт-Петербург: Питер, 2022. — 896 с. — ISBN 978-5-4461-1408-5.

Интернет ресурсы:

1. Документация Microsoft .NET. — URL: https://learn.microsoft.com/ru-ru/dotnet/ (дата обращения: ..20__). — Текст: электронный.
2. Документация по C#. — URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/ (дата обращения: ..20__). — Текст: электронный.
3. Документация Entity Framework Core. — URL: https://learn.microsoft.com/ru-ru/ef/core/ (дата обращения: ..20__). — Текст: электронный.
4. Портал сообщества .NET (dotnet.ru). — URL: https://dotnet.ru/ (дата обращения: ..20__). — Текст: электронный.

13.3 Кадровое обеспечение программы

Реализация образовательной программы профессионального обучения обеспечивается педагогическими кадрами, имеющими высшее образование/среднее профессиональное образование, соответствующее профилю.

14. Контроль и оценка результатов освоения программы

Контроль и оценка результатов освоения программы включает текущий контроль знаний, промежуточную и итоговую аттестацию обучающихся.

Текущий контроль успеваемости является систематической проверкой получаемых знаний и практических навыков обучающимися. Текущий контроль успеваемости проводится преподавателем на любом из видов учебных занятий. Формы и методы текущего контроля успеваемости выбираются преподавателем исходя из специфики программы, формируемых общих и профессиональных компетенций и устанавливаются рабочей программой.

Промежуточная аттестация является процедурой оценки степени и уровня освоения обучающимися отдельной части или всего объема программы. Формы и порядок промежуточной аттестации выбираются образовательной организацией самостоятельно, периодичность промежуточной аттестации определяется учебным планом.

Итоговая аттестация проводится в целях определения соответствия результатов освоения обучающимися образовательный программы. Итоговая аттестация проводится в форме зачета.

Оценочные материалы

Общие сведения

Контроль и оценка результатов освоения рабочей программы осуществляется преподавателями в процессе проведения практических занятий, а также сдачи слушателями итоговой аттестации в форме зачета.

В качестве форм и методов контроля и оценки результатов обучения, позволяющих проверить сформированность профессиональных и общих компетенций, осуществляется:

– текущий контроль.

Текущий контроль представляет собой проверку усвоения учебного материала, регулярно осуществляемую на протяжении обучения. Текущая проверка позволяет оценить совокупность знаний и умений и формирование компетенций, соответствующих специальности. Текущий контроль осуществляется в формах: собеседование, выполнение типовых заданий для практических работ.

– экспертная оценка результатов наблюдений за деятельностью слушателей в процессе освоения программы.

– выпускная итоговая работа.

Осуществляется после освоения программы дополнительного профессионального образования в полном объёме. Выпускная итоговая работа включает выполнение практического задания.

Условия проведения

Зачет проводится в форме выполнения практического задания

Количество вариантов задания: 2

Оборудование:

Указывается перечень используемого оборудования

– Visual Studio 2022, MS SQL Server Express

Критерии и система оценивания

Критерии оценки:

По результатам итоговой аттестации выставляется оценка «зачет» или «незачет»

Система оценивания:

Выполнение заданий оценивается по одному критерию:

– выполнение практического задания;

Оценка	Условие
Зачёт	Часть 1 (консольное приложение): создана иерархия классов (не менее 2 классов), реализованы LINQ-запросы (не менее 2 видов операций), работает сохранение в JSON/БД. Часть 2 (оконное приложение): форма содержит DataGridView и элементы управления, реализовано добавление и удаление записей, данные загружаются из БД через EF Core. Теоретический тест: не более 3 ошибок
Незачёт	Часть 1: классы не созданы или нарушены принципы ООП, LINQ-запросы не работают, сохранение данных не реализовано. Часть 2: форма не загружается или отсутствуют обработчики событий, подключение к БД не выполнено. Теоретический тест: более 3 ошибок

Перечень вопросов/заданий для подготовки к итоговой аттестации:

Задания для итоговой зачетной работы (по вариантам)

Вариант 1. Мини-проект «Каталог товаров»

Часть 1. Консольное приложение

Задания:

1. Создайте базовый класс Product с полями: Id, Name, Price, Category. Реализуйте конструктор и свойства.
2. Создайте производный класс DiscountedProduct, добавив поле Discount (в процентах) и переопределив свойство Price для расчёта цены со скидкой.
3. Создайте коллекцию List<Product> из 8 товаров разных категорий («Электроника», «Одежда», «Продукты»).
4. Напишите LINQ-запросы:

– Вывести товары категории «Электроника» с ценой выше 5000 руб., отсортированные по цене.

– Вывести среднюю цену товаров каждой категории.

– Вывести топ-3 самых дешёвых товара.

5. Реализуйте сохранение коллекции в JSON-файл и загрузку из файла с обработкой исключений.

Часть 2. Оконное приложение (Windows Forms)

Задания:

6. Создайте форму «Каталог товаров» с DataGridView, полями ввода (Name, Price, Category, Discount) и кнопками «Добавить», «Удалить выбранный», «Сохранить».
7. Подключите базу данных через Entity Framework Core (Code First). Создайте контекст и выполните миграцию.
8. Реализуйте загрузку данных из БД в DataGridView при запуске приложения.
9. Реализуйте обработчик кнопки «Добавить»: создание нового товара, добавление в БД и обновление DataGridView.
10. Реализуйте обработчик кнопки «Удалить выбранный»: удаление товара из БД и обновление DataGridView.

Вариант 2. Мини-проект «Учёт сотрудников»

Часть 1. Консольное приложение

Задания:

1. Создайте базовый класс Employee с полями: Id, FullName, Position, Salary. Реализуйте конструктор и свойства.
2. Создайте производный класс Manager, добавив поле Department и переопределив метод GetInfo() для вывода полной информации.
3. Создайте коллекцию List<Employee> из 8 сотрудников разных должностей.
4. Напишите LINQ-запросы:

– Вывести сотрудников с зарплатой выше среднего значения.

– Вывести количество сотрудников в каждой должности.

– Вывести топ-5 сотрудников по зарплате.

5. Реализуйте сохранение коллекции в JSON-файл и загрузку из файла с обработкой исключений.

Часть 2. Оконное приложение (Windows Forms)

Задания:

6. Создайте форму «Учёт сотрудников» с DataGridView, полями ввода (FullName, Position, Salary, Department для Manager) и кнопками «Добавить», «Удалить выбранного», «Сохранить».

7. Подключите базу данных через Entity Framework Core (Code First). Создайте контекст и выполните миграцию.
8. Реализуйте загрузку данных из БД в DataGridView при запуске приложения.
9. Реализуйте обработчик кнопки «Добавить»: создание нового сотрудника, добавление в БД и обновление DataGridView.
10. Реализуйте обработчик кнопки «Удалить выбранного»: удаление сотрудника из БД и обновление DataGridView.

Методические материалы

Методические материалы в помощь обучающимся предназначены для эффективного освоения дополнительной профессиональной программы.

Рекомендации по выполнению вариантов итоговой аттестационной работы

Итоговая аттестация проводится в форме зачета и включает выполнение практического задания по одному из двух вариантов: **Вариант 1 «Каталог товаров»** или **Вариант 2 «Учёт сотрудников»**. Задание состоит из двух частей: консольное приложение и оконное приложение (Windows Forms) с подключением базы данных.

Общие рекомендации:

1. **Внимательно прочитайте задание.** В каждой части перечислены конкретные шаги. Выполняйте их последовательно — результат каждого шага используется в последующих.
2. **Следите за именованием.** Используйте осмысленные названия классов, методов и переменных в стиле PascalCase (классы, методы) и camelCase (переменные). Например: Product, GetDiscountedPrice(), productName.
3. **Проверяйте работоспособность после каждого шага.** Запустите приложение и убедитесь, что добавленный код работает корректно, прежде чем переходить к следующему шагу.

Рекомендации к Части 1 (консольное приложение):

- Создание базового класса. Начните с определения полей и свойств. Используйте автосвойства ({ get; set; }) для простых случаев. Конструктор должен принимать все обязательные параметры.
- Создание производного класса. При наследовании используйте синтаксис class DerivedClass : BaseClass. Для переопределения свойства или метода используйте ключевые слова virtual в базовом классе и override в производном.
- Работа с коллекцией. Используйте List<T> для хранения объектов. Заполните коллекцию с помощью инициализатора коллекции или метода Add().
- Написание LINQ-запросов. Используйте два синтаксиса на выбор: «метод-цепочек» (.Where(), .OrderBy(), .GroupBy(), .Select()) или «выражений запросов» (ключевые слова from, where, orderby, group, select). Для вывода результатов используйте цикл foreach.
- Сохранение в JSON. Используйте System.Text.Json.JsonSerializer.Serialize() для преобразования коллекции в строку и File.WriteAllText() для записи в файл. Для загрузки — File.ReadAllText() и JsonSerializer.Deserialize<T>(). Оберните файловые операции в блок try-catch для обработки ошибок (файл не найден, недостаточно прав и т.д.).

Рекомендации к Части 2 (оконное приложение с Entity Framework Core):

- Создание формы. Используйте Windows Forms (.NET Framework или .NET Core). Разместите на форме: DataGridView для отображения таблицы, TextBox-ы для ввода данных, Button-ы для действий («Добавить», «Удалить выбранный», «Сохранить»).
- Настройка DataGridView. Установите свойства AutoSizeColumnsMode = Fill, SelectionMode = FullRowSelect, AllowUserToAddRows = False. Запретите редактирование напрямую в гриде — все операции только через кнопки.
- Подключение Entity Framework Core. Установите NuGet-пакеты: Microsoft.EntityFrameworkCore.SqlServer и Microsoft.EntityFrameworkCore.Tools. Создайте класс контекста БД, унаследованный от DbContext, с набором DbSet<T>. В методе OnConfiguring() укажите строку подключения к MS SQL Server.
- Миграция. Откройте «Консоль диспетчера пакетов» и выполните команды: Add-Migration InitialCreate, затем Update-Database. Убедитесь, что таблицы создались в БД.
- Асинхронная загрузка данных. Используйте async/await при обращении к БД: await context.Products.ToListAsync(). Это предотвратит зависание интерфейса.

– Обработчики кнопок. При добавлении записи: создайте новый объект из данных в полях ввода, добавьте через context.Add(), вызовите await context.SaveChangesAsync() и обновите DataGridView. При удалении: получите выбранную строку через dataGridView.SelectedRows, определите Id, найдите объект в БД, удалите через context.Remove() и сохраните изменения.

Дополнительные рекомендации:

– Для проверки содержимого БД используйте «Обозреватель объектов SQL Server» (Server Explorer) в Visual Studio.

– При возникновении ошибок обращайте внимание на сообщения компилятора и среды выполнения — они содержат указание на место и причину ошибки.

– Перед итоговой аттестацией повторите основные LINQ-операторы, синтаксис CRUD-запросов EF Core и работу с DataGridView.

– Для подготовки используйте официальную документацию Microsoft .NET и материалы курса.