

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Ректор Белорусского
государственного университета

А.Д.Король

27 июня 2025 г.

Регистрационный № 2814/б.



ЯЗЫК ПРОГРАММИРОВАНИЯ C#. ОСНОВЫ ASP.NET

Учебная программа учреждения образования по учебной дисциплине для
специальности:

6-05-0533-07 Математика и компьютерные науки

Профилизация: Веб-программирование и интернет-технологии

2025 г.

Учебная программа составлена на основе образовательного стандарта ОСВО 6-05-0533-07-2023, учебных планов: № 6-5.4-55/03 от 15.05.2023, № 6-5.4-55/13 ин. от 31.05.2023, № 6-54-55-23з. от 31.05.2023.

СОСТАВИТЕЛЬ:

А.И.Кравчук, доцент кафедры веб-технологий и компьютерного моделирования механико-математического факультета Белорусского государственного университета, кандидат физико-математических наук, доцент.

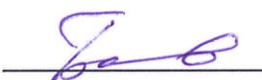
РЕЦЕНЗЕНТ:

А.С. Кравчук, профессор кафедры экономической информатики учреждения образования «Белорусский государственный экономический университет», доктор физико-математических наук, доцент.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой веб-технологий и компьютерного моделирования БГУ
(протокол № 17 от 02.06.2025);

Научно-методическим советом БГУ
(протокол № 11 от 26.06.2025)

Декан механико-математического факультета  С.М.Босяков

 (С.М.Босяков)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

.NET представляет собой программную платформу, выпущенную компанией Microsoft в 2002 году, для создания приложений на базе семейства операционных систем Windows, а также некоторых операционных систем производства не Microsoft, таких как Mac OS X и различные дистрибутивы Unix и Linux. Платформа .NET

- обеспечивает возможность взаимодействия с существующим программным кодом;
- позволяет поддерживать многочисленные языки программирования, приоритетным из которых является язык программирования C#;
- имеет общий исполняющий механизм, используемый всеми поддерживаемыми .NET языками;
- характеризуется полной интеграцией языков – поддерживается межязыковое наследование, обработка исключений и отладка кода;
- имеет обширную библиотеку базовых классов, позволяющую избегать сложностей, связанных с выполнением прямых вызовов к API-интерфейсу, и предлагает согласованную объектную модель, которую могут использовать все поддерживаемые .NET языки.

Наличие перечисленных функциональных возможностей делает платформу .NET совместно с языком программирования C# сверхмощным инструментом для создания различного рода приложений. Разработчики программного обеспечения на основе платформы .NET в настоящий момент являются востребованными, что делает платформу и язык C# актуальными для изучения студентами.

Изучение дисциплины «Язык программирования C#. Основы ASP.NET» базируется на знаниях, полученных при изучении университетских курсов «Методы программирования и информатика», «Web-программирование», «Базы данных».

Цели и задачи учебной дисциплины

Дисциплина «Язык программирования C#. Основы ASP.NET» имеет прикладную направленность.

Целью дисциплины «Язык программирования C#. Основы ASP.NET» является изучение языка, базовых технологий .NET Framework, а также технологии разработки веб-приложений с использованием новейших версий фреймворка ASP.NET.

Задачи учебной дисциплины:

- создавать приложения на языке программирования C#;
- строить архитектуру разрабатываемого приложения в соответствии с принципами ООП и SOLID;
- пользоваться сторонними программами и инструментами в процессе разработки реальных программных продуктов;

- настраивать взаимодействие Web-сайтов с базами данных;
- проектировать и разрабатывать web-системы с использованием технологий ASP.NET;

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина «Язык программирования C#. Основы ASP.NET» относится к **модулю** «Веб-программирование» компонента учреждения образования.

Связи с другими учебными дисциплинами, включая учебные дисциплины компонента учреждения высшего образования, дисциплины специализации и др.

Изучение дисциплины «Язык программирования C#. Основы ASP.NET» взаимосвязана с учебной дисциплиной «Разработка веб-приложений на ASP.NET MVC».

Требования к компетенциям

Освоение учебной дисциплины «Язык программирования C#. Основы ASP.NET» должно обеспечить формирование следующих компетенций:

специализированные компетенции:

Выполнять проектирование, разработку, тестирование, и маркетинг информационных решений в сети Интернет с учетом их последующего масштабирования и обработки возникающих больших объемов данных.

В результате изучения дисциплины студент должен:

знать:

- синтаксис языка C#;
- основные концепции платформы .NET и ее базовой части Framework;
- базовые принципы разработки web-приложений на основе технологии ASP.NET;

уметь:

- создавать и вызывать методы;
- перехватывать, обрабатывать и выбрасывать исключения;
- создавать и использовать новые типы (перечисления, классы, структуры, интерфейсы, делегаты;
- выполнять простые операции для взаимодействия с файловой системой;
- контролировать область видимости и время жизни членов типа;
- использовать наследование для разработки новых ссылочных типов данных;
- управлять временем жизни объектов и контролировать использование ресурсов;
- определять свойства и индексы для инкапсулирования данных;

- отделять действие от метода, реализующего это действие, и использовать отделенные операции для обработки асинхронных событий;
 - использовать коллекции для объединения данных и использовать обобщения для реализации типизированных классов, структур, интерфейсов и методов;
 - реализовывать собственные классы коллекций, поддерживающие возможность перебора элементов;
 - создавать запросы к данным, находящимся в памяти, с использованием LINQ;
 - выполнять первичное тестирование проекта на этапах разработки;
 - планировать и внедрять архитектуру проекта в соответствии с принципами ООП;
 - создавать веб-приложения в интегрированной среде разработки и развертывания Microsoft Visual Studio на языке C#;
 - создавать веб-системы на основе фреймворка ASP.NET MVC, реализующий шаблон «Модель-Представление-Контроллер»;
 - решать задачи, требующие доступа к данным, используя LINQ;
 - управлять состоянием в веб-приложениях;
- иметь навык:**
- работы в среде разработки Microsoft Visual Studio 2019;
 - разработки web-приложений в среде Microsoft Visual Studio 2019.

Структура учебной дисциплины

Дисциплина изучается в 5 семестре. В соответствии с учебным планом всего на изучение учебной дисциплины «Язык программирования C#. Основы ASP.NET» отведено для **очной формы** получения высшего образования – 90 часов, в том числе 36 аудиторных часов: лекции – 18 часов, лабораторные занятия – 18 часов.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма промежуточной аттестации – зачет.

Дисциплина изучается в 7 семестре. В соответствии с учебным планом всего на изучение учебной дисциплины «Язык программирования C#. Основы ASP.NET» отведено для **заочной формы** получения высшего образования – 90 часов, в том числе 8 аудиторных часов, из них: лекции – 4 часа, лабораторные занятия – 4 часа.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма промежуточной аттестации – зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Введение в .NET Framework. Синтаксис и основные программные конструкции языка C#.

Знакомство с платформой .NET Framework и ее возможностями. Цели и преимущества платформы при разработке приложений на C#. Работа в Visual Studio: создание проектов, запуск, отладка. Введение в базовый синтаксис языка C# и его ключевые конструкции. Примеры консольных и графических приложений. Использование инструментов анализа: ILDASM, .NET Reflector. Работа с типами данных и операциями, массивами, условиями и циклами.

Тема 2. Классы и объекты в C#.

Создание и использование собственных классов. Объявление полей, методов и свойств. Разработка и вызов конструкторов. Передача параметров: обязательные, необязательные, выходные. Методы с переменным числом аргументов. Основы объектно-ориентированного подхода.

Тема 3. Инкапсуляция данных и методов.

Инкапсуляция как принцип ООП. Использование модификаторов доступа. Статические члены класса и их особенности. Статические конструкторы. Индексаторы для доступа по индексу. Перегрузка операторов и реализация преобразований типов.

Тема 4. Наследование, абстракция и полиморфизм.

Механизм наследования в C#. Работа с интерфейсами и абстрактными классами. Реализация виртуальных и переопределяемых методов. Сокращение дублирования кода за счет наследования. Объявление и использование структур и перечислений.

Тема 5. Исключения и обработка ошибок.

Обработка ошибок в приложениях. Использование try-catch-finally. Встроенные классы исключений и создание пользовательских. Механизмы генерации исключений. Эффективное управление ошибками на уровне приложения.

Тема 6. Работа с файловой системой.

Чтение и запись данных в файлы. Работа с текстовыми и бинарными файлами. Использование пространства имен System.IO. Применение потоков. Разбор различных форматов. Организация доступа к файловой системе.

Тема 7. Делегаты и события.

Использование делегатов для создания ссылок на методы. Обработка событий в приложении. Применение анонимных методов и лямбда-выражений. Принципы реактивного программирования и обратной связи.

Тема 8. Коллекции и обобщенные типы. LINQ.

Работа с обобщенными коллекциями и интерфейсами. Использование обобщенных методов и делегатов. Создание пользовательских коллекций. Применение LINQ для запросов к данным. Сравнение статических и динамических LINQ-запросов.

Тема 9. Введение в ASP.NET Core.

Знакомство с платформой ASP.NET Core и ее назначением. Отличия от ASP.NET Framework. Основные принципы архитектуры. Кроссплатформенность и модульность. Поддержка разных типов приложений. Структура базового проекта ASP.NET Core.

Тема 10. Создание и запуск первого приложения.

Создание проекта в Visual Studio с использованием шаблонов. Обзор структуры проекта. Основные файлы: Program.cs, appsettings.json. Работа со средами окружения. Конфигурация и запуск приложения.

Тема 11. Конвейер обработки запросов и Middleware.

Жизненный цикл HTTP-запроса. Использование встроенных Middleware. Создание и регистрация пользовательских Middleware. Порядок выполнения компонентов. Взаимодействие Middleware и логика обработки запроса.

Тема 12. Класс Startup и конфигурация приложения.

Работа с IConfiguration. Использование конфигурационных файлов appsettings.json и переменных среды. Переопределение настроек в зависимости от окружения. Подключение внешних источников конфигурации.

Тема 13. Внедрение зависимостей (Dependency Injection).

Регистрация сервисов с различным временем жизни: Transient, Scoped, Singleton. Использование интерфейсов. Внедрение зависимостей в компоненты. Использование встроенного контейнера DI в ASP.NET Core.

Тема 14. Обработка HTTP-запросов.

Использование MapGet, MapPost и других методов маршрутизации. Работа с параметрами маршрута, строки запроса и тела запроса. Возврат JSON-ответов. Работа с объектом HttpContext.

Тема 15. Создание Web API.

Реализация REST API без использования MVC. Обработка HTTP-методов GET, POST, PUT, DELETE. Получение и валидация входных данных. Формирование структурированных ответов. Использование TypedResults.

Тема 16. Аутентификация и авторизация.

Настройка аутентификации через куки и JWT. Определение политик доступа. Авторизация по ролям. Защита маршрутов. Использование системы Identity и настройка безопасности на уровне приложения.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная (дневная) форма получения высшего образования с применением дистанционных образовательных технологий (ДОТ)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Формы контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	Введение в .NET Framework. Синтаксис и основные программные конструкции языка C#	2			1			Защита лабораторных работ
2	Классы и объекты	1			1			Защита лабораторных работ
3	Инкапсуляция данных и методов в C#	1			1			Защита лабораторных работ
4	Наследование в C#. Интерфейсы и абстрактные классы	1			1			Опрос
5	Исключения и обработка исключений	1			1			Защита лабораторных работ
6	Работа с файловой системой	1			1			Защита лабораторных работ
7	Делегаты и события	1			1			Защита лабораторных работ
8	Коллекции и обобщенные типы. Введение в LINQ	1			1			Защита лабораторных работ
9	Введение в ASP.NET Core	1			1			Защита лабораторных работ

10	Создание и запуск первого приложения	1				1			Защита лабораторных работ
11	Конвейер обработки запросов и Middleware	1				1			Защита лабораторных работ
12	Класс Startur и конфигурация приложения	1				1			Опрос
13	Внедрение зависимостей (Dependency Injection)	1				1			Защита лабораторных работ
14	Обработка HTTP-запросов	1				1			Защита лабораторных работ
15	Создание Web API	2				1	2		Защита лабораторных работ
16	Аутентификация и авторизация	1				1			Защита лабораторных работ
	ВСЕГО: 36 часов	18				16	2		Зачет

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Заочная форма получения высшего образования

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов VCP	Формы контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1.	Введение в .NET Framework. Синтаксис и основные программные конструкции языка C#	0.25			0.25			Опрос
2.	Классы и объекты	0.25			0.25			Опрос
3.	Инкапсуляция данных и методов в C#	0.25			0.25			Опрос
4.	Наследование в C#. Интерфейсы и абстрактные классы	0.25			0.25			Опрос
5.	Исключения и обработка исключений	0.25			0.25			Опрос
6.	Работа с файловой системой	0.25			0.25			Опрос
7.	Делегаты и события	0.25			0.25			Опрос
8.	Коллекции и обобщенные типы. Введение в LINQ	0.25			0.25			Опрос
9.	Введение в ASP.NET Core	0.25			0.25			Опрос
10.	Создание и запуск первого	0.25			0.25			Опрос

	приложения											
11.	Конвейер обработки запросов и Middleware	0.25					0.25					Опрос
12.	Класс Startup и конфигурация приложения	0.25					0.25					Опрос
13.	Внедрение зависимостей (Dependency Injection)	0.25					0.25					Опрос
14.	Обработка HTTP-запросов	0.25					0.25					Опрос
15.	Создание Web API	0.25					0.25					Опрос
16.	Аутентификация и авторизация	0.25					0.25					Опрос
	ВСЕГО: 8 часов	4					4					Зачет

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература

1. Прайс, Марк. C# 9 и .NET 5. Разработка и оптимизация / Марк Прайс; [пер. с англ. С. Черников]. – 5-е изд. – Санкт-Петербург ; Москва ; Минск : Питер, 2022. – 832 с. – URL: <https://ibooks.ru/bookshelf/382387>.
2. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Джеффри Рихтер; [пер. с англ. Е. Матвеев]. – 4-е изд. – Санкт-Петербург ; Москва; Минск : Питер, 2024. – 895 с. – URL: <https://ibooks.ru/reading.php?short=1&productid=376829>.
3. Маран, М. М. Программная инженерия : учебное пособие / М. М. Маран. – Изд. 3-е, стер. – Санкт-Петербург ; Москва ; Краснодар : Лань, 2022. – 194 с. – URL: <https://reader.lanbook.com/book/189470#2>.
4. Симан, Марк. Внедрение зависимостей на платформе .NET = Dependency Injection / Марк Симан, Стивен Ван Дерсен ; [пер. с англ. Н. Вильчинского]. – 2-е изд. – Санкт-Петербург [и др.] : Питер, 2021. – 605 с. – URL: <https://ibooks.ru/bookshelf/376970>.

Дополнительная литература

1. Joseph Albahari. C# 12 in a Nutshell: The Definitive Reference. O'Reilly Media, 2023 – 1083 p.
2. Adam Freeman. Pro ASP.NET Core 7, Tenth Edition. Manning Publications, 2023 – 1256 стр.
3. Andrew Lock. ASP.NET Core in Action, Third Edition. Manning Publications, 2023 – 800 p.
4. <https://metanit.com/sharp/tutorial/>
5. <https://metanit.com/sharp/aspnet6/>

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Объектом диагностики компетенций студентов являются знания, умения, полученные ими в результате изучения учебной дисциплины. Выявление учебных достижений студентов осуществляется с помощью мероприятий текущей и промежуточной аттестации.

Для текущего контроля качества усвоения знаний по дисциплине рекомендуется использовать:

- опрос;
- отчеты по лабораторным с их устной защитой.

Формой промежуточной аттестации по дисциплине «Язык программирования C#. Основы ASP.NET» учебным планом предусмотрен зачет.

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема 15. Создание Web API. (2 часа)

Разработать Web API-приложение на ASP.NET Core, реализующее функциональность управления списком книг (или других сущностей по выбору преподавателя). API должно поддерживать базовые операции CRUD (Create, Read, Update, Delete).

Функциональные требования:

- Создание проекта:
 - Использовать шаблон ASP.NET Core Web API (без MVC).
 - Удалить лишние примеры из шаблона, оставив только нужные файлы и настройки.
- Создание модели данных:
 - Определить C#-класс Book (или другую сущность), содержащий как минимум следующие поля:
 - Id (int)
 - Title (string)
 - Author (string)
 - Year (int)
- Реализация CRUD-операций через HTTP-методы:
 - GET /books — получение списка всех книг
 - GET /books/{id} — получение одной книги по ID
 - POST /books — добавление новой книги
 - PUT /books/{id} — обновление существующей книги
 - DELETE /books/{id} — удаление книги
- Хранение данных:
 - Использовать в качестве временного хранилища встроенный список в памяти (List<Book>).
- Ответы API:
 - Возвращать корректные коды состояния (200, 201, 404, 400 и др.)
 - Использовать JSON-сериализацию
- Документация и тестирование:
 - Добавить Swagger (Swashbuckle) для интерактивной документации

API

- Убедиться, что все операции работают через Swagger UI

Ожидаемый результат:

- Функционирующее Web API с корректной маршрутизацией, сериализацией и CRUD-операциями.
- Опрятная структура проекта.
- Понимание жизненного цикла HTTP-запросов в ASP.NET Core.
- Навыки возврата правильных ответов и обработки ошибок.
- Умение документировать API через Swagger.

Форма контроля опрос, отчет по лабораторным заданиям.

Примерный перечень лабораторных занятий

Тема 1. Введение в .NET Framework.

Установить Visual Studio и .NET SDK. Создать консольное приложение на C#. Ознакомиться с файлами проекта. Изучить сборку .exe с помощью ILDASM или .NET Reflector.

Тема 2. Классы и объекты.

Реализовать класс с несколькими полями, методами и свойствами. Написать программу, создающую и использующую объекты этого класса.

Тема 3. Инкапсуляция.

Создать класс с закрытыми полями, доступ к которым осуществляется через свойства. Добавить статические члены и индексатор. Продемонстрировать перегрузку операторов.

Тема 4. Наследование и абстракция.

Реализовать иерархию классов с использованием базового, абстрактного класса и интерфейса. Показать применение полиморфизма.

Тема 5. Обработка исключений.

Создать программу, в которой обрабатываются различные исключения. Добавить пользовательский класс исключения и использовать его.

Тема 6. Работа с файлами.

Реализовать запись и чтение текстовой информации в файл. Дополнительно: реализовать сохранение данных в бинарном формате.

Тема 7. Делегаты и события.

Создать делегат и связать его с методами. Реализовать простую модель событий — например, уведомление при достижении порога значения.

Тема 8. Обобщения и LINQ.

Создать обобщенный класс (например, стек). Использовать LINQ-запросы для обработки списка объектов (фильтрация, сортировка, выборка).

Тема 9. Введение в ASP.NET Core.

Изучить структуру и назначение компонентов ASP.NET Core. Создать пустой проект и разобрать его содержимое.

Тема 10. Структура проекта.

Создать новое приложение на ASP.NET Core с шаблоном "Empty". Ознакомиться с файлами Program.cs, appsettings.json. Настроить логирование.

Тема 11. Middleware.

Добавить в проект несколько Middleware-компонентов. Реализовать пользовательский Middleware для логирования времени обработки запроса.

Тема 12. Конфигурация.

Добавить различные настройки в appsettings.json. Прочитать их в коде через IConfiguration. Создать структуру для привязки конфигурации.

Тема 13. Dependency Injection.

Зарегистрировать и внедрить простой сервис в приложение. Реализовать интерфейс и конкретную реализацию, используемую в Middleware.

Тема 14. Обработка HTTP-запросов.

Настроить обработку маршрутов с помощью MapGet и MapPost. Передавать параметры через URL и тело запроса. Возвращать простые JSON-ответы.

Тема 15. Web API.

Реализовать Web API с методами GET и POST. Использовать типизированные ответы (TypedResults) и обрабатывать входные данные с проверкой.

Тема 16. Аутентификация и авторизация.

Ознакомиться с базовой настройкой аутентификации. Настроить доступ к маршрутам с использованием RequireAuthorization() и ролей (опционально).

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса используются

- **методы и приемы развития критического мышления**, которые представляют собой систему, формирующую навыки работы с информацией в процессе чтения и письма; понимания информации как отправного, а не конечного пункта критического мышления;

- **метод группового обучения**, который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями

- **практико-ориентированный подход**, который предполагает освоение содержание образования через решения практических задач.

Методические рекомендации по организации самостоятельной работы обучающихся

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

- изучение литературы и материалов электронных источников по проблемам дисциплины;
- выполнение домашнего задания;
- подготовка к лабораторным занятиям;
- курсовые, дипломные и научно-исследовательские работы, связанные с тематикой дисциплины;
- подготовка к участию в конференциях с докладами по проблемам дисциплины.

Примерный перечень вопросов к зачету

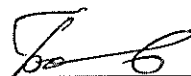
1. Что такое .NET Framework и какова его архитектура?
2. Как устроены сборки в .NET и что такое GAC?
3. Основные типы данных в C#. Примеры использования.
4. Как осуществляется приведение типов в C#?
5. Какие существуют управляющие конструкции в языке C#?
6. Что такое массивы и как они объявляются и инициализируются?
7. Что такое класс в C#? Как определить поля, методы и свойства?
8. Как работают конструкторы в C# и какие их виды существуют?
9. Что такое инкапсуляция? Для чего нужны модификаторы доступа?
10. Что означает ключевое слово `static` и где оно используется?
11. Что такое наследование и как его реализовать в C#?
12. В чем различия между интерфейсом и абстрактным классом?
13. Как реализуется полиморфизм в C#?
14. Как устроена система обработки исключений?
15. Как создать пользовательское исключение?
16. Что такое делегат? Примеры использования.
17. Что такое событие в C# и как оно генерируется и обрабатывается?
18. Что такое лямбда-выражения и чем они удобны?
19. Что такое обобщения? Какие преимущества они дают?
20. Что такое LINQ? Примеры запросов и их применение.
21. Что такое ASP.NET Core и чем он отличается от ASP.NET Framework?
22. Какие компоненты входят в архитектуру ASP.NET Core-приложения?
23. Как создать простое веб-приложение на ASP.NET Core?
24. Что такое Middleware и как они работают в конвейере запросов?
25. Как осуществляется конфигурация приложения в ASP.NET Core?
26. Какие источники конфигурации поддерживаются?
27. Что такое внедрение зависимостей и как оно реализовано в ASP.NET Core?
28. В чем различия между Transient, Scoped и Singleton сервисами?
29. Как обрабатываются HTTP-запросы в минималистичной модели ASP.NET Core?
30. Как определить маршрут и обработчик с помощью MapGet / MapPost?
31. Как создать простое Web API без контроллеров MVC?
32. Как происходит работа с параметрами и телом запроса в Web API?
33. Какие существуют способы возврата ответов в API?
34. Что такое аутентификация и авторизация в ASP.NET Core?
35. Какие подходы используются для реализации безопасности в ASP.NET Core?

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Учебная дисциплина не требует согласования			

Декан механико-математического факультета

д.ф.-м.н., профессор
02.06.2025



С.М.Босяков

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ УО

на ____/____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры
_____ (протокол № ____ от _____ 202_ г.)

Заведующий кафедрой

УТВЕРЖДАЮ

Декан факультета
