

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Ректор Белорусского
государственного университета

А.Д.Король

15 июля 2024 г.

Регистрационный №УД- 13401/уч.



РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЙ НА ASP.NET MVC

Учебная программа учреждения образования
по учебной дисциплине для специальности:

**1-31 03 08 Математика и информационные технологии
(по направлениям)**

Направления специальности:

1-31 03 08-01 Веб-программирование и интернет-технологии

2024 г.

Учебная программа составлена на основе ОСВО 1-31 03 08-2021 и учебных планов №G31-1-011/уч. от 25.05.2021, №G31-1-003/уч.ин. от 31.05.2021, №G31-1-004/уч.3 от 31.05.2021, №G31-1-220/уч. от 22.03.2022, №G31-1-225 уч.ин. от 27.05.2022 №G31-1-218/ч.3. от 27.05.2022.

СОСТАВИТЕЛЬ:

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

РЕЦЕНЗЕНТ:

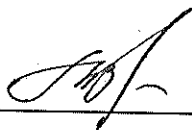
А.С. Кравчук, профессор кафедры экономической информатики Белорусского государственного экономического университета, доктор физ.-мат. наук, доцент.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой веб-технологий и компьютерного моделирования БГУ
(протокол № 19 от 05.06.2024);

Научно-методическим советом БГУ
(протокол № 9 от 28.06.2024)

Заведующий кафедрой



М.В. Игнатенко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ASP.NET MVC – это мощный и популярный фреймворк для разработки веб-приложений на платформе .NET, который сочетает простоту и гибкость. Он позволяет создавать современные, масштабируемые, и надежные веб-приложения, отвечающие требованиям бизнеса.

Фреймворк ASP.NET MVC:

- Поддерживает шаблон проектирования MVC (Model-View-Controller), который способствует разделению логики приложения, интерфейса и данных;
- Предоставляет мощные инструменты для создания веб-интерфейсов, включая интеграцию с HTML, CSS и JavaScript;
- Включает поддержку RESTful сервисов и API;
- Позволяет использовать сторонние библиотеки и фреймворки, такие как Entity Framework, для работы с данными;
- Интегрируется с современными DevOps-процессами, включая контейнеризацию и CI/CD.

Программа по учебной дисциплине «Разработка веб-приложений на ASP.NET MVC» относится к дисциплинам специализации и разработана для студентов 4 курса очной формы получения высшего образования, для студентов 5 курса заочной формы обучения получения высшего образования обучающихся по специальностям 1-31 03 08 Математика и информационные технологии (по направлениям) направления специальности 1-31 03 08-01 Математика и информационные технологии (веб-программирование и интернет-технологии).

Цели и задачи учебной дисциплины

Дисциплина «Разработка веб-приложений на ASP.NET MVC» имеет прикладную направленность.

Целью дисциплины является освоить разработку веб-приложений на платформе .NET с использованием архитектурного подхода ASP.NET MVC, современных технологий и инструментов.

Задачи учебной дисциплины

- Изучить архитектурные принципы ASP.NET MVC, включая раздельное управление логикой, интерфейсом и данными.
- Освоить методы проектирования и реализации моделей данных с использованием Entity Framework Core и взаимодействия с базами данных.
- Изучить подходы к созданию пользовательских интерфейсов (Views), включая применение Razor, HTML, CSS и JavaScript для динамической генерации контента.
- Развить практические навыки разработки контроллеров, настройки маршрутизации (Routing) и применения фильтров для обработки запросов.
- Изучить основы разработки RESTful API на основе ASP.NET MVC для обмена данными между клиентом и сервером.

- Освоить инструменты обеспечения безопасности, включая аутентификацию, авторизацию, защиту от XSS и CSRF-атак.
- Научиться эффективно использовать современные инструменты разработки, такие как Visual Studio, Git и браузеры для тестирования.
- Получить навыки оптимизации производительности веб-приложений и работы с кэшированием.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина «Разработка веб-приложений на ASP.NET MVC» относится к дисциплинам специализаций компонента учреждения высшего образования.

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

Изучение дисциплины «Разработка веб-приложений на ASP.NET MVC» базируется на знаниях, полученных при изучении дисциплин «Язык программирования C#. Основы ASP.NET», «Веб-программирование».

Требования к компетенциям

Освоение учебной дисциплины «Разработка веб-приложений на ASP.NET MVC» должно обеспечить формирование следующих компетенций:

специализированные компетенции:

СК. 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:

- Основные концепции языка C# и платформы .NET.
- Принципы проектирования веб-приложений на ASP.NET MVC.
- Основы взаимодействия веб-приложений с базами данных с использованием LINQ и Entity Framework Core.

- Способы интеграции внешних API и сервисов.

уметь:

- Проектировать, разрабатывать и тестировать веб-приложения, соблюдая архитектуру MVC.
- Настраивать маршрутизацию, контроллеры и модели для работы с различными источниками данных.

владеть:

- Навыками разработки серверной части веб-приложений на языке C# с использованием ASP.NET MVC.
- Инструментами для повышения производительности приложений, включая кэширование, асинхронные операции и оптимизацию запросов.
- Методами обеспечения безопасности веб-приложений (аутентификация, авторизация).

Структура учебной дисциплины

Дисциплина изучается в 7 семестре дневной и 9 семестре заочной формы получения высшего образования. Всего на изучение учебной дисциплины «Разработка веб-приложений на ASP.NET MVC» отведено:

– для очной формы получения высшего образования – 90 часов, в том числе 36 аудиторных часов, из них: лекции – 18 часов, лабораторные занятия – 16 часов, управляемая самостоятельная работа (УСР) – 2 часа.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма промежуточной аттестации – зачет.

– 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 ASP.NET MVC

Архитектурные принципы MVC: модель, представление, контроллер. Платформа .NET: возможности и преимущества для веб-разработки. Основные инструменты разработки: Visual Studio, Git, браузеры для тестирования. Обзор технологий: Razor, Entity Framework, LINQ.

Тема 2. Настройка среды разработки

Установка и настройка Visual Studio 2022. Создание первого проекта ASP.NET MVC. Структура проекта: папки, файлы, конфигурация. Настройка маршрутизации (Routing).

Тема 3. Разработка моделей (Model)

Принципы создания моделей данных. Использование Entity Framework Core для работы с базами данных. Настройка подключения к базе данных. Миграции базы данных.

Тема 4. Контроллеры (Controller)

Основы работы с контроллерами: создание и настройка. Маршрутизация запросов: атрибуты и параметры. 4.3. Обработка данных и бизнес-логика в контроллерах. Асинхронные методы в контроллерах.

Тема 5. Представления (View)

Создание и настройка представлений. Использование Razor для динамического формирования HTML. Работа с ViewBag, ViewData и Model в представлениях. Частичные представления (Partial Views).

Тема 6. Взаимодействие с базой данных

Запросы к базе данных с использованием LINQ. Работа с репозиториями и Unit of Work. CRUD-операции: создание, чтение, обновление, удаление данных. Оптимизация запросов: кэширование и индексирование.

Тема 7. Работа с формами и вводом данных

Создание форм в ASP.NET MVC. Валидация данных на стороне клиента и сервера. Работа с HTML-хелперами для упрощения создания форм. Обработка ошибок при вводе данных.

Тема 8. Безопасность веб-приложений

Аутентификация и авторизация пользователей. Защита от XSS, CSRF и SQL-инъекций. Управление ролями и правами доступа. Настройка HTTPS и шифрования.

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 УСП	Формы контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1.	Введение в ASP.NET MVC	2			2			Опрос
2.	Настройка среды разработки	2			2			Опрос, отчет по лабораторным заданиям
3.	Разработка моделей (Model)	2			2			Опрос, отчет по лабораторным заданиям
4.	Контроллеры (Controller)	2			2			Опрос, отчет по лабораторным заданиям
5.	Представления (View)	2			2			Опрос, отчет по лабораторным заданиям
6.	Взаимодействие с базой данных	4			2		2	Опрос, отчет по лабораторным заданиям
7.	Работа с формами и вводом данных	2			2			Опрос, отчет по лабораторным заданиям
8.	Безопасность веб-приложений	2			2			Опрос, отчет по лабораторным заданиям
	ВСЕГО: 36 часов	18			16		2	Зачёт. Защита итогового проекта

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 ASP.NET MVC	0.5			0.5			Опрос
2.	Настройка среды разработки	0.5			0.5			Опрос
3.	Разработка моделей (Model)	0.5			0.5			Опрос
4.	Контроллеры (Controller)	0.5			0.5			Опрос
5.	Представления (View)	0.5			0.5			Опрос
6.	Взаимодействие с базой данных	0.5			0.5			Опрос
7.	Работа с формами и вводом данных	0.5			0.5			Опрос
8.	Безопасность веб-приложений	0.5			0.5			Опрос
	ВСЕГО: 8 часов	4			4			Зачёт

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература

1. Полное руководство по ASP.NET Core и платформе .NET. – URL: <https://metanit.com/sharp/aspnetmvc/>
2. Adam Freeman: Pro ASP.NET Core 6. Publisher: Apress, 2022. – 1253 p.
3. ASP.NET Core documentation. Microsoft Documentation – URL: <https://learn.microsoft.com/ru-ru/aspnet/core/>.

Дополнительная литература

1. Andrew Lock: ASP.NET Core in Action, Third Edition. Publisher: Manning Publications, 2023. – 984 p.

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Диагностика результатов учебной деятельности по дисциплине «Разработка веб-приложений на ASP.NET MVC» проводится в рамках аудиторных занятий и самостоятельной работы студентов. Для диагностики используются следующие методы:

- устный опрос;
- отчет по лабораторным заданиям;
- защита итогового проекта.

Отметка за ответы на лекциях (опрос) и лабораторных занятиях включает следующие критерии: полнота и глубина ответа, наличие аргументов, использование примеров из реальной практики.

Оценка отчета по лабораторным заданиям и итогового проекта проводится с учетом следующих параметров:

- актуальность и сложность поставленных задач;
- корректность применяемых методов решения;
- качество выполнения задания (работа с кодом, оформление, тестирование);
- привлечение знаний из смежных областей;
- использование современных инструментов и технологий.

Контроль самостоятельной работы студентов (УСР) осуществляется преподавателем с применением современных средств ИКТ. Это может быть проведение онлайн-тестирования, проверка загруженных заданий через систему управления обучением, а также обсуждение решений в формате консультаций.

Полученные студентом количественные результаты включаются в состав итоговой отметки по дисциплине в рамках рейтинговой системы.

Формой промежуточной аттестации по дисциплине «Разработка веб-приложений на ASP.NET MVC» учебным планом предусмотрен зачет.

Для формирования итоговой отметки по учебной дисциплине используется модульно-рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов для текущей и промежуточной аттестации студентов по учебной дисциплине.

Формирование итоговой отметки в ходе проведения контрольных мероприятий текущей аттестации (примерные весовые коэффициенты, определяющие вклад текущей аттестации в отметку при прохождении промежуточной аттестации):

- устный опрос – 10%;
- отчеты по лабораторным работам – 40%;
- защита итогового проекта – 50%.

Итоговая отметка по дисциплине рассчитывается на основе итоговой отметки текущей аттестации (рейтинговой системы оценки знаний) 40 % и экзаменационной отметки 60 %.

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема 6. Взаимодействие с базой данных (2 ч.)

– Подключение к базе данных. Настроить подключение к базе данных, содержащей информацию о клиентах: таблица Clients (ID, Name, Email, Phone, Address).

– Получение данных по клиенту. Создать программу, которая получает информацию о клиенте по его ID (например, имя, контактные данные).

– Добавление нового клиента. Реализовать метод для добавления нового клиента в таблицу Clients (поля: Name, Email, Phone, Address).

– Обновление данных клиента. Реализовать метод для обновления информации о клиенте, например, изменение адреса или номера телефона.

– Удаление клиента из базы данных. Реализовать метод для удаления клиента из таблицы по ID.

– Создание хранимой процедуры для получения списка клиентов. Написать хранимую процедуру для выборки всех клиентов с возможностью фильтрации по имени или email (например, список клиентов с именем, начинающимся на "Анна").

– Вызов хранимой процедуры из программы. Реализовать метод, который вызывает хранимую процедуру для получения списка клиентов по фильтру (например, по части имени) и выводит результат в консоль.

Форма контроля: Опрос, отчет по лабораторным заданиям

Примерный перечень тем итогового проекта

1. Интернет-магазин

Разработать веб-приложение для управления каталогом товаров, корзиной покупок и оформлением заказов.

Основные функции: регистрация пользователей, авторизация, управление категориями и товарами, поиск товаров, добавление товаров в корзину, оформление и отслеживание заказов.

2. Система управления персоналом (HR-система)

Веб-приложение для учета сотрудников компании и автоматизации кадровых процессов.

Основные функции: управление списком сотрудников, создание профилей сотрудников, учёт рабочего времени, генерация отчётов по отделам и должностям.

3. Платформа для онлайн-курсов

Приложение для публикации, просмотра и прохождения онлайн-курсов.

Основные функции: регистрация студентов и преподавателей, создание курсов и уроков, тестирование студентов, выдача сертификатов по результатам.

4. Система бронирования

Приложение для бронирования услуг, например, номеров в отелях, столиков в ресторанах или билетов на мероприятия.

Основные функции: поиск доступных услуг, выбор даты и времени, оплата бронирования, подтверждение и уведомления.

5. Трекер задач (Task Tracker)

Веб-приложение для управления задачами и проектами в команде.

Основные функции: создание проектов, добавление задач, назначение задач пользователям, статус задач, комментарии и файлы к задачам.

6. Мессенджер для внутреннего общения

Приложение для обмена сообщениями внутри компании или команды.

Основные функции: регистрация пользователей, личные и групповые чаты, отправка файлов, уведомления о новых сообщениях.

7. Система управления библиотекой

Приложение для учета книг в библиотеке и их выдачи пользователям.

Основные функции: добавление книг в каталог, поиск книг, бронирование и выдача книг, учёт задолженностей.

8. Система управления событиями (Event Management)

Приложение для создания, организации и отслеживания событий, таких как конференции или встречи.

Основные функции: регистрация участников, создание расписания, уведомления о предстоящих событиях, генерация отчетов о посещаемости.

9. Портал новостей

Веб-приложение для публикации и управления новостями.

Основные функции: создание категорий, публикация статей, добавление комментариев, фильтрация и поиск новостей, авторизация для редакторов.

10. Финансовый трекер

Приложение для учёта личных финансов или бюджета компании.

Основные функции: добавление доходов и расходов, создание категорий для операций, генерация финансовых отчетов, интеграция с банковскими системами.

11. Социальная сеть для профессионалов

Приложение для создания профессиональных профилей, публикации достижений и поиска вакансий.

Основные функции: регистрация пользователей, создание профилей, поиск коллег, публикация объявлений о вакансиях, система рекомендаций.

12. Платформа для бронирования спортивных площадок

Приложение для аренды спортивных залов и площадок.

Основные функции: выбор площадки, просмотр расписания, бронирование и оплата.

13. Система учёта студентов

Веб-приложение для управления списком студентов и их успеваемостью.

Основные функции: регистрация студентов, учёт успеваемости, создание расписания занятий, автоматизация генерации отчетов.

14. Портал вопросов и ответов (Q&A)

Платформа для создания и обсуждения вопросов.

Основные функции: публикация вопросов, добавление ответов, голосование за лучший ответ, модерирование контента.

15. Сервис доставки еды

Приложение для заказа еды из ресторанов.

Основные функции: каталог блюд, оформление заказов, управление статусами доставки, система отзывов и рейтингов.

Каждый проект должен соответствовать архитектуре ASP.NET MVC, включать функциональные модули для работы с базой данных, реализацию маршрутизации, использование моделей, контроллеров и представлений. Проект должен быть разработан с учётом принципов безопасности, иметь продуманный пользовательский интерфейс, обработку ошибок и предоставлять документацию, описывающую основные функции и их использование.

Описание итогового проекта

Цель: Создать функциональное веб-приложение, демонстрирующее применение архитектурной модели MVC, работу с базой данных, аутентификацию пользователей и основные механизмы ASP.NET MVC.

Описание проекта:

Студентам предлагается разработать веб-приложение, имитирующее работу системы управления, интернет-магазина или другой предметной области на выбор (например, система бронирования, блог, трекер задач). Проект должен быть выполнен на платформе ASP.NET MVC с использованием технологий Razor, Entity Framework Core и LINQ.

Требования к проекту:

Архитектура и структура приложения

- Использование архитектуры MVC (разделение логики на модели, представления и контроллеры).
- Четкая организация проекта, включающая описание структуры папок, маршрутов и зависимостей.

Работа с базой данных

- Проект должен использовать Entity Framework Core для работы с базой данных.
- Реализация как минимум двух связанных таблиц с использованием миграций.
- CRUD-операции (создание, чтение, обновление и удаление данных) для основных сущностей.

Функциональность приложения

- Реализация страниц для отображения списка данных, создания, редактирования и удаления записей.
- Использование форм и их валидация на стороне клиента и сервера.
- Использование частичных представлений для повторяющихся элементов интерфейса.

Аутентификация и авторизация

Реализация системы регистрации и входа пользователей.

Настройка ролей и прав доступа для различных категорий пользователей.

Дизайн и интерфейс

- Использование Razor для динамического формирования HTML.
- Опционально: интеграция Bootstrap или другого CSS-фреймворка для улучшения внешнего вида.

Дополнительные функции:

- Кэширование данных для повышения производительности.
- Фильтрация и сортировка данных в таблицах.
- Отчёты или аналитика с использованием LINQ-запросов.

Этапы выполнения:

Проектирование:

- Определение предметной области.
- Разработка схемы базы данных.
- Составление списка функциональных требований.

Разработка:

- Создание проекта и настройка среды разработки.
- Реализация базового функционала (работа с базой данных, маршрутизация, CRUD).
- Добавление пользовательского интерфейса и валидации данных.

Тестирование:

- Проверка работы маршрутов и форм.
- Тестирование функционала регистрации и авторизации.

- Поиск и исправление ошибок.

Презентация:

- Демонстрация работы приложения.
- Объяснение реализации ключевых функций и технических решений.

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса используются:

- **Методы проектного обучения**, которые обеспечивают студентам возможность интегрировать теоретические знания и практические навыки через разработку реальных проектов. Это способствует развитию самостоятельности, креативности и ответственности за результат.
- **Практико-ориентированный подход**, предполагающий освоение учебного материала через решение задач, максимально приближенных к реальным профессиональным ситуациям. Такой подход формирует компетенции, необходимые для работы в IT-индустрии.
- **Применение современных технологий обучения**, включая использование онлайн-ресурсов, виртуальных лабораторий и платформ для совместной работы, таких как GitHub. Это помогает студентам освоить инструменты, которые востребованы в профессиональной деятельности.
- **Методы формирования soft skills**, включающие работу в командах, навыки презентации проектов и развитие критического мышления. Эти методы обеспечивают подготовку студентов не только как технических специалистов, но и как эффективных участников команды.
- **Игровые технологии и симуляции**, например, создание мини-проектов, моделирующих реальные процессы в веб-разработке. Это позволяет студентам получить практический опыт в условиях, приближенных к производственным.
- **Обратная связь в реальном времени**, которая включает регулярные обсуждения результатов работы студентов, анализ типичных ошибок и рекомендации по их исправлению. Такой подход повышает эффективность обучения и мотивацию.

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.

Для организации самостоятельной работы студентов по учебной дисциплине используются современные информационные ресурсы, размещенные на образовательном портале смешанного и дистанционного обучения БГУ, содержащие учебные материалы (курс лекций, задания к лабораторным и домашним работам и т. д.)

Примерный перечень вопросов к зачету

1. Архитектурные принципы MVC: модель, представление, контроллер.
2. Возможности платформы .NET для веб-разработки.
3. Основные инструменты разработки ASP.NET MVC: Visual Studio, Git.
4. Настройка среды разработки и создание первого проекта ASP.NET MVC.
5. Структура проекта ASP.NET MVC: папки, файлы, конфигурация.
6. Основы маршрутизации (Routing): настройка и использование.
7. Принципы создания моделей данных в ASP.NET MVC.
8. Использование Entity Framework Core для работы с базами данных.
9. Настройка подключения к базе данных и использование миграций.
10. Создание и настройка контроллеров в ASP.NET MVC.
11. Маршрутизация запросов: параметры, атрибуты.
12. Асинхронные методы в контроллерах.
13. Создание представлений и работа с Razor.
14. Использование ViewBag, ViewData и моделей в представлениях.
15. Работа с частичными представлениями (Partial Views).
16. Выполнение запросов к базе данных с использованием LINQ.
17. Реализация CRUD-операций (создание, чтение, обновление, удаление данных).
18. Использование репозитория и паттерна Unit of Work.
19. Создание форм в ASP.NET MVC и работа с HTML-хелперами.
20. Валидация данных на стороне клиента и сервера.
21. Обработка ошибок при вводе данных и уведомление пользователей.
22. Основы безопасности веб-приложений: аутентификация и авторизация.
23. Принципы кэширования и оптимизация запросов к базе данных.
24. Организация структуры проекта и его компонентов.
25. Логирующее и обработка ошибок в ASP.NET MVC.
26. Основные этапы проектирования и разработки веб-приложения.
27. Презентация и документирование итогового проекта.
28. Лучшие практики и рекомендации для разработки веб-приложений.

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 требует согласования			

Заведующий кафедрой

канд. физ.-мат. н., доцент
(ученая степень, ученое звание)


(подпись)

И. В. Урманяко
(И.О. Фамилия)

05. 06 2024 г.

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ

на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры

_____ (название кафедры) (протокол № ____ от _____ 202_ г.)

Заведующий кафедрой

_____ (ученая степень, ученое звание)

_____ (подпись)

_____ (И.О. Фамилия)

УТВЕРЖДАЮ
Декан факультета

_____ (ученая степень, ученое звание)

_____ (подпись)

_____ (И.О. Фамилия)