

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

О.Г. Прохоренко

«05» июля 2023 г.

Регистрационный №УД - 12164/уч.



Язык программирования C#. Основы ASP.NET

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальности:**

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

Направление специальности:

**1-31 03 08-01 Математика и информационные технологии
(Веб-программирование и интернет-технологии)**

2023 г.

Программа разработана на основе ОСВО 1-31 03 08-2021, типового учебного плана, регистрационный № G31-1-012/тип. от 31.03.2021 по специальности 1-31 03 08 Математика и информационные технологии (по направлениям) и учебных планов БГУ: по направлению специальности 1-31 03 08-01 Математика и информационные технологии (веб-программирование и интернет-технологии) № G31-1-011/ уч. от 25.05.2021 г., № G31-1-004/ уч.з от 31.05.2021 г., № G31-1-003/ уч.ин. от 31.05.2021 г., № G31-1-220/ уч. от 22.03.2022 г., № G31-1-218/ уч.з от 27.05.2022 г., № G31-1-225/ уч.ин. от 27.05.2022 г.

СОСТАВИТЕЛЬ:

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

РЕЦЕНЗЕНТ:

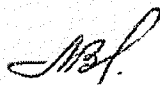
А.М. Зеневич, заведующая кафедрой экономической информатики Белорусского государственного экономического университета, кандидат экономических наук, доцент.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой веб-технологий и компьютерного моделирования БГУ
(протокол № 9 от 24.05.2023 г.)

Научно-методическим Советом БГУ
(протокол № 9 от 29.06.2023 г.)

Заведующий кафедрой



М.В. Игнатенко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

.NET – это платформа с открытым исходным кодом для создания настольных, мобильных и веб-приложений, которые могут работать в любой операционной системе. 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 интеграций языков – поддерживается межъязыковое 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 поддерживающие .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 является востребованными, что делают платформу и язык C# актуальными для изучения студентами.

Учебная дисциплина «Язык программирования C#. Основы ASP.NET» предназначена для студентов специальности 1-31 03 08 Математика и информационные технологии (по направлениям) направление специальности: 1-31 03 08-01 Математика и информационные технологии (Веб-программирование и интернет-технологии).

Цели и задачи учебной дисциплины

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

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

Задачами учебной дисциплины дисциплины «Язык программирования C#. Основы ASP.NET» являются научить студентов:

- 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 с базами данных;
- проектировать и разрабатывать web-системы с использованием технологии ASP.NET Core;

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина «Язык программирования C#. Основы ASP.NET» относится к дисциплинам специализаций 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» базируется на знаниях, полученных при изучении дисциплин «Методы программирования», «Веб-программирование», «Базы данных».

Требования к компетенциям

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

Базовые профессиональные компетенции:

БПК – 6. Применять современные технологии и базовые конструкции языков программирования для реализации алгоритмических прикладных задач и разработки веб-проектов.

Специализированные компетенции:

СК – 9. 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 BCL;
- 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 Core;
- владеть:**
- навыками работы в среде разработки Microsoft Visual Studio 2022;
 - навыками разработки web-приложений в среде Microsoft Visual Studio 2022.

Структура учебной дисциплины

Дисциплина изучается в 5 семестре дневной формы получения высшего образования и в 8-9-семестре заочной формы получения высшего образования. 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, из них: лекции – 18 часов, лабораторные занятия – 16 часов, управляемая самостоятельная работа – 2 часа.
- для заочной формы получения высшего образования – 90 часов, в том числе 10 аудиторных часов, из них: лекции – 6 часов, лабораторные занятия – 4 часа.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы дневной формы получения высшего образования и 1 зачетную единицу заочной формы получения высшего образования.

Форма промежуточной аттестации - зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Язык C# и основы платформы .NET.

Тема 1.1. Введение в .NET. Синтаксис и основные программные конструкции языка C#. Цели .NET и возможности, доступные при создании приложений с помощью C# и Visual Studio. Знакомство с платформой .NET новейшей версии. Создание проектов с Visual Studio новейшей версии. Типы проектов. Написание приложений на C#. Отладка исходного кода, документирование приложения. Основные типы данных и программные конструкции языка C#, их синтаксис и семантику. Приведение типов. Литералы. Объявление переменных и присваивание значений. Использование выражений и операций языка в C#. Использование операторов выбора, цикла, условного оператора. Массивы. Создание и использование массивов.

Тема 1.2. Классы и объекты в C#. Синтаксис объявления классов. Конструкторы. Поля и методы, их определение и использование. Спецификация необязательных и выходных параметров, методы с переменным числом параметров. Свойства, их объявление и использование.

Тема 1.3. Инкапсуляция данных и методов. Модификаторы доступа в C# и реализация принципов инкапсуляции. Модификатор static и его использование. Статические конструкторы. Индексаторы. Перегрузка операторов. Операторы преобразования типов.

Тема 1.4. Наследование в C#. Интерфейсы и абстрактные классы. Наследование классов. Виртуальные методы и полиморфизм. Интерфейсы и абстрактные классы, а также возможности их использования для упрощения сложных проблем сокращения дублирования кода и ускорения разработки. Создание и использование структур и перечислений.

Тема 1.5. Исключения и обработка исключений. Работа с файловой системой. Знакомство с исключениями и их классами. Как можно реализовывать в приложениях их эффективную обработку. Обработка исключений. Возникновение исключений. Пользовательские исключения.

Тема 1.6. Делегаты и события. Объявление и использование делегатов. Ковариация и контравариация делегатов. События и их обработка. Как использовать события для информирования приложений об изменении или заметном явлении в типе. Лямбда-выражения. Анонимные методы.

Тема 1.7. Коллекции и обобщенные типы. Введение в LINQ. Обобщения и возможности их использования для поддержания целостности типа и избегания проблем, связанных с отсутствием безопасности типов базовых классов коллекций. Знакомство с обобщенными интерфейсами и классами коллекциями .NET, и возможностью разработки пользовательских классов коллекций, поддерживающих конструкции языка C#, обеспечивающих перечисление и инициализацию. Использование коллекций. Создание и использование обобщенных типов. Определение обобщенных интерфейсов и понимание вариативности. Использование обобщенных методов и делегатов. Знакомство с

LINQ (Language-Integrated Query) запросами и возможностями их использования для обработки данных в приложениях .NET.

Раздел 2. ASP.NET Core MVC

Тема 2.1. Введение в ASP.NET Core. Клиент-серверное взаимодействие. Общие концепции ASP.NET Core. Компоненты приложения ASP.NET Core. Модель компиляции ASP.NET Core. Структура приложений ASP.NET Core.

Тема 2.2. MVC паттерн. Фреймворк ASP.NET Core MVC. Базовые концепции ASP.NET Core MVC.

Тема 2.3. Маршрутизация. Определение маршрутов. Входящая и исходящая маршрутизация. Атрибуты маршрутизации.

Тема 2.4. Контроллеры и методы действий. Знакомство с классом-контроллером. Получение входных данных. Создание выходных данных.

Тема 2.5. Представления. Механизм представления Razor. Работа с движком представления Razor. Добавление динамического контента в представление Razor. Вспомогательные методы, шаблонные вспомогательные методы. Создание пользовательских вспомогательных методов. Использование встроенных вспомогательных методов. Использование шаблонных вспомогательных методов. Использование метаданных модели.

Тема 2.6. Фильтры. Использование фильтров. Использование фильтров авторизации. Использование фильтров исключений. Использование фильтров методов действий. Использование фильтров результата действия. Использование встроенных фильтров.

Тема 2.7. Связывание данных модели. Понимание модели связывания данных. Использование модели связывания данных по умолчанию. Вызов модели связывания данных вручную. Настройка системы модели связывания данных.

Тема 2.8. Валидация модели. Явная валидация модели. Отображение сообщений валидации. Использование альтернативных методов валидации. Выполнение валидации на стороне клиента. Выполнение удаленной валидации.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

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

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов VCP	Формы контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	Язык C# и основы платформы .NET							
1.1	Введение в .NET. Синтаксис и основные программные конструкции языка C#	1			2			Опрос, отчет по лабораторным / домашним заданиям
1.2	Классы и объекты	1			1			Опрос, отчет по лабораторным / домашним заданиям
1.3	Инкапсуляция данных и методов в C#	1			1			Опрос, отчет по лабораторным / домашним заданиям
1.4	Наследование в C#. Интерфейсы и абстрактные классы.	2			1			Опрос, отчет по лабораторным / домашним заданиям
1.5	Исключения и обработка исключений. Работа с файловой системой.	1			1			Опрос, отчет по лабораторным / домашним заданиям
1.6	Делегаты и события.	1			1			Опрос, отчет по лабораторным / домашним заданиям
1.7	Коллекции и обобщенные типы. Введение в LINQ	2			1			Опрос, отчет по лабораторным / домашним заданиям
2	ASP.NET Core MVC							
2.1	Введение в ASP.NET Core	1			1			Опрос, отчет по лабораторным / домашним заданиям
2.2	MVC паттерн. Фреймворк ASP.NET Core MVC	1			1			Опрос, отчет по лабораторным / домашним заданиям

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов VCP	Формы контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
2.3	Маршрутизация	1			1			Опрос, отчет по лабораторным / домашним заданиям
2.4	Контроллеры и методы действий	2			1		2	Опрос, отчет по лабораторным / домашним заданиям
2.5	Представления	1			1			Опрос, отчет по лабораторным / домашним заданиям
2.6	Фильтры.	1			1			Опрос, отчет по лабораторным / домашним заданиям
2.7	Связывание данных модели	1			1			Опрос, отчет по лабораторным / домашним заданиям
2.8	Валидация модели	1			1			Опрос, отчет по лабораторным / домашним заданиям
	ВСЕГО: 36 часов	18			16		2	Зачёт

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Заочная форма получения высшего образования

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Формы контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	Язык С# и основы платформы .NET	3			2			Опрос, отчет по лабораторным / домашним заданиям
2	ASP.NET Core MVC	3			2			Опрос, отчет по лабораторным / домашним заданиям
	ВСЕГО: 10 часов	6			4			Зачёт

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. Руководство по ASP.NET Core 7
2. ASP.NET documentation
3. Полное руководство по языку программирования C# 11.0 и платформе .NET 7.
4. 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. : ил. ; 23х16 см. - (Серия "Для профессионалов") (Packt) (Expert Insight). - Загл. ориг.: C# 9 and .NET 5 – Modern Cross-Platform Development. - ISBN 978-5-4461-2921-8. - ISBN 978-1800568105.
5. Руководство по Entity Framework Core 7

Перечень дополнительной литературы

6. ASP.NET documentation
7. Руководство по Entity Framework Core 7

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Диагностика результатов учебной деятельности по дисциплине «Язык программирования С#. Основы ASP.NET» проводится, как правило, во время аудиторных занятий. Для диагностики используются:

- устный опрос;
- отчет по лабораторным и домашним заданиям.

Оценка за ответы на лекциях (опрос) и лабораторных занятиях включает в себя полноту ответа, наличие аргументов, примеров из практики и т. д.

Оценка отчета по лабораторным и домашним заданиям включает актуальность исследуемой проблемы, корректность используемых методов исследования, привлечение знаний из сопредельных областей, организация работы группы.

Контроль УСП проводится преподавателем с использованием ИКТ в форме опроса и проверки результатов выполнения лабораторных и домашних работ.

Полученные студентом количественные результаты учитываются как составная часть итоговой отметки по дисциплине в рамках рейтинговой системы.

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

При формировании итоговой отметки используется рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов в ходе проведения контрольных мероприятий текущей аттестации.

Примерные весовые коэффициенты, определяющие вклад текущей аттестации в отметку при прохождении промежуточной аттестации:

Формирование отметки за текущую аттестацию:

- устный опрос – 20%;
- отчёты по лабораторным работам – 40%;
- отчёты по домашним работам – 40%.

Итоговая отметка по дисциплине рассчитывается на основе отметки текущей аттестации и отметки на зачёте с учетом их весовых коэффициентов. Вес отметки по текущей аттестации составляет 40%, отметки на зачёте – 60%.

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема 2.4. Контроллеры и методы действия. (2 ч)

Задание 1. Создание контроллера и действий для управления списком задач.

1. Создайте новый контроллер "TaskController", который будет управлять списком задач.

В контроллере "TaskController" создайте следующие действия:

- "Index" - отображает список всех задач.
- "Create" - позволяет создать новую задачу.
- "Edit" - позволяет редактировать существующую задачу.
- "Delete" - позволяет удалить задачу.

2. Создайте модель "Task" для представления данных о задачах. Модель "Task" должна включать поля, такие как "ID", "Название задачи", "Описание", "Дата создания" и другие, в зависимости от вашего проекта.

3. Реализуйте представления для действий "Create", "Edit" и "Index", которые позволят пользователям создавать, редактировать и просматривать задачи соответственно.

4. Используйте Entity Framework Core для взаимодействия с базой данных и сохранения задач.

5. На странице "Index" отобразите список всех задач, предоставив пользователю возможность просматривать, редактировать и удалять их.

6. Добавьте маршруты, чтобы контроллер "TaskController" был доступен по URL-адресам, таким как "/Task" для списка задач, "/Task/Create" для создания новой задачи и так далее.

7. Добавьте валидацию данных для полей задач, чтобы предотвратить сохранение некорректных данных.

Задание позволит студентам практиковаться в создании контроллеров, действий, моделей и представлений, а также взаимодействовать с базой данных и реализовывать основные операции CRUD (создание, чтение, обновление, удаление) для задач.

Форма контроля - опрос, отчет по лабораторным и домашним заданиям.

Задание 2. Создание контроллера и действий для управления списком продуктов в интернет-магазине.

1. Создайте новый контроллер "ProductController", который будет управлять списком продуктов в интернет-магазине.

2. В контроллере "ProductController" создайте следующие действия:

- "Index" - отображает список всех продуктов.
- "Create" - позволяет администратору добавлять новые продукты в магазин.
- "Edit" - позволяет администратору редактировать существующие продукты.
- "Delete" - позволяет администратору удалять продукты из каталога.

3. Создайте модель "Product" для представления данных о продуктах. Модель "Product" должна включать поля, такие как "ID", "Наименование продукта", "Цена", "Описание", "Количество в наличии" и другие, в зависимости от вашего проекта.
 4. Реализуйте представления для действий "Create", "Edit" и "Index", которые позволят администратору магазина создавать, редактировать и просматривать информацию о продуктах.
 5. Используйте Entity Framework Core для взаимодействия с базой данных и сохранения информации о продуктах.
 6. На странице "Index" отобразите список всех продуктов, предоставив администратору возможность управлять ими.
 7. Добавьте маршруты, чтобы контроллер "ProductController" был доступен по URL-адресам, таким как "/Product" для списка продуктов, "/Product/Create" для добавления нового продукта и так далее.
 8. Реализуйте валидацию данных для полей продуктов, чтобы предотвратить сохранение некорректных данных, например, отрицательных цен или нулевого количества продуктов в наличии.
- Это задание позволит студентам практиковаться в создании контроллеров, действий, моделей и представлений, а также взаимодействовать с базой данных и управлять информацией о продуктах в интернет-магазине.
- Форма контроля** - опрос, отчет по лабораторным и домашним заданиям.

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;
- **практико-ориентированный подход**, который предполагает освоение содержания 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.

Для организации самостоятельной работы студентов по учебной дисциплине рекомендуется использовать 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. Объявление переменных и присваивание значений.
2. Использование выражений и операций языка в C#.
3. Использование операторов выбора.
4. Использование операторов цикла.
5. Создание и использование массивов.
6. Определение и вызов методов.
7. Спецификация необязательных и выходных параметров.
8. Взаимодействие типов, объектов, стека потока и управляемой кучи во время выполнения.
9. Создание и использование перечислений.
10. Создание и использование классов.
11. Создание и использование структур.
12. Сравнение ссылочных типов с типами значений.
13. Управление видимостью членов типа.
14. Свойства и индексаторы.
15. Статические методы и данные.
16. Перегрузка операций.
17. Использование наследования для определения новых ссылочных типов.
18. Определение и реализация интерфейсов.
19. Определение абстрактных классов.
20. Обработка исключений.
21. Возникновение исключений.
22. Управление ресурсами.
23. Доступ к файловой системе.
24. Чтение и запись файлов с помощью потоков.
25. Объявление и использование делегатов.
26. Использование лямбда-выражений.
27. Обработка событий.
28. Использование коллекций.
29. Создание и использование обобщенных типов.
30. Определение обобщенных интерфейсов.
31. Использование обобщенных методов и делегатов.
32. Реализация пользовательского класса коллекции.
33. ASP.NET Core и его отличия от более ранних версий ASP.NET.

34. Структура проекта ASP.NET Core MVC и файлы модели, представления и контроллера.
35. Преимущества ASP.NET Core MVC по сравнению с другими платформами.
36. Маршрутизация в ASP.NET Core MVC и настройка маршрутов.
37. Роль модели в архитектуре ASP.NET Core MVC и виды моделей.
38. Контроллеры в ASP.NET Core MVC и их обработка запросов.
39. Методы действий в контроллерах для обработки HTTP-запросов.
40. Передача данных между контроллером и представлением в ASP.NET Core MVC.
41. Razor View Engine и его использование для создания представлений.
42. Атрибуты маршрутизации и их влияние на обработку запросов.
43. Middleware в ASP.NET Core и его связь с MVC.
44. Валидация данных и инструменты безопасности в ASP.NET Core MVC.
45. Использование Dependency Injection в ASP.NET Core MVC и его важность.
46. Обработка исключений в ASP.NET Core MVC и методы обработки ошибок.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

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)

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
Веб-технологий и компьютерного моделирования (протокол № ____
от _____ 202_ г.)

Заведующий кафедрой
Кандидат физ. мат. наук, доцент

М.В. Игнатенко

УТВЕРЖДАЮ
Декан факультета
Доктор физ. мат. наук, доцент

С.М. Босяков