

А.М. Минитаева

Разработка приложений на языке C#: создание веб-серверов и backend-разработка

Учебно-методическое пособие



Москва

ИЗДАТЕЛЬСТВО

МГТУ им. Н.Э. Баумана

2024

УДК 681.3.06
ББК 22.18
М57

Издание доступно в электронном виде по адресу
<https://press.bmstu.ru/catalog/item/8184/>

Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»

*Рекомендовано Научно-методическим советом
МГТУ им. Н.Э. Баумана в качестве учебно-методического пособия*

Рецензент
канд. техн. наук, доцент *С.Р. Муминова*

Минитаева, А. М.

М57 Разработка приложений на языке C#: создание веб-серверов и backend-разработка : учебно-методическое пособие / А. М. Минитаева. — Москва : Издательство МГТУ им. Н. Э. Баумана, 2024. — 36, [4] с. : ил.

ISBN 978-5-7038-6303-9

Приведены основные теоретические сведения о языке C#, разработке консольных приложений, backend-компонентах информационных систем, веб-серверах и TCP-соединениях, необходимые для выполнения домашних заданий № 1 и 2 по созданию HTTP-серверов и backend-приложений на языке C#. Рассмотрены примеры.

Для студентов 4-го курса МГТУ им. Н.Э. Баумана, обучающихся по направлениям подготовки бакалавриата «Информатика и вычислительная техника» и «Прикладная информатика» и изучающих дисциплину «Разработка приложений на языке C#».

УДК 681.3.06
ББК 22.18



Уважаемые читатели! Пожелания, предложения, а также сообщения о замеченных опечатках и неточностях Издательство просит направлять по электронной почте: press@bmstu.ru

ISBN 978-5-7038-6303-9

© МГТУ им. Н.Э. Баумана, 2024
© Оформление. Издательство
МГТУ им. Н.Э. Баумана, 2024

Предисловие

Учебно-методическое пособие посвящено принципам создания веб-серверов и backend-разработкам на языке C# в среде Visual Studio. Оно должно оказать помощь обучающимся в освоении технологии кросс-платформенной разработки программного обеспечения и при выработке практических навыков создания веб-серверов и backend-компонентов программных продуктов.

В языке программирования C# (созданном в компании Microsoft для поддержки собственной среды .NET Framework) проверенные временем средства языков C и C++ усовершенствованы с помощью самых современных технологий для решения полного круга задач, стоящих на сегодняшний день перед разработчиками. Язык C# и среда разработки Visual Studio представляют собой очень эффективный способ создания программного обеспечения для современной инфраструктуры вычислительной обработки данных, которая включает в себя требования кросс-платформенности, операционные системы Windows, Linux, Unix, Android и др., глобальную сеть Internet и локальные вычислительные сети, различные архитектуры процессоров и микроконтроллеров. Освоение языка программирования C# и среды разработки Visual Studio способствует расширению профессиональных возможностей студентов, получающих подготовку по некоторым IT-специальностям.

Цель издания — оказать помощь студентам при приобретении ими навыков создания веб-серверов и backend-разработки на языке C#, необходимых для выполнения домашнего задания по дисциплине «Разработка приложений на языке C#».

Пособие включает в себя домашние задания № 1 и № 2 в соответствии с модулями 1 и 2 учебной программы по дисциплине «Разработка приложений на языке C#».

Домашнее задание № 2 состоит из частей 1 и 2, посвященных созданию статического и динамического веб-приложений. Кроме того, его часть 2 предусматривает использование полученных результатов в части 1 и домашнем задании № 1.

Выполнение представленных домашних заданий позволит студентам приобрести навыки самостоятельного создания проектов на языке C# в среде Visual Studio и технологичной разработки приложений, предназначенных для вычислительных и backend-компонентов современных информационных систем, а также ознакомиться с синтаксисом языка программирования C#.

Домашние задания № 1 и № 2 содержат теоретическую и практическую части. В теоретической части приведены необходимые инструкции, пояснения и примеры, в практической — непосредственно задание, подробные методы создания вычислительного программного обеспечения для практического использования при разработке консольных приложений, описание процесса разработки HTTP-сервера и методики создания веб-серверов и backend-приложений, обеспечивающих взаимодействие с клиентом посредством JSON-запросов. К каждому домашнему заданию приведены вопросы и задания для закрепления полученных знаний.

Для изучения материала пособия обязательно предварительное освоение дисциплины «Информатика» с помощью учебного пособия «Кодирование информации. Системы счисления. Основы логики»*.

* Кодирование информации. Системы счисления. Основы логики : учеб. пособие / А.М. Минитаева. М.: Изд-во МГТУ им. Н. Э. Баумана, 2019. 106 с.

ДОМАШНЕЕ ЗАДАНИЕ № 1

Консольный калькулятор арифметических выражений

Цель домашнего задания — развитие навыков программирования на языке C#, создание консольного приложения, способного вычислять арифметические выражения.

Домашнее задание предусматривает выполнение следующих задач:

- разработка алгоритма для парсинга (обработки строки) введенного пользователем арифметического выражения;
- реализация парсера, который будет преобразовывать строку с арифметическим выражением в последовательность операций;
- разработка алгоритма для вычисления значения арифметического выражения на основе последовательности операций;
- реализация калькулятора, который будет принимать арифметические выражения от пользователя, парсить их, вычислять и выводить результаты;
- разработка тестов для проверки корректности работы калькулятора.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Разработка простых приложений

Наиболее простым видом приложений являются консольные приложения. Консольное приложение (или консольный интерфейс) — это приложение, которое запускается в командной строке (консоли) операционной системы (ОС) и взаимодействует с пользователем через текстовый интерфейс. В консольном приложении пользователь вводит команды с помощью клавиатуры и видит вывод информации на экране монитора. Для того чтобы создать консольное приложение, необходимо сообщить среде разработки Visual Studio, что приложение будет использовать подсистему консоли.

Консольные приложения отлично подходят для выполнения рутинных задач и автоматизации различных процессов. Они могут быть написаны на языке программирования C# и выполнять широкий спектр задач — от обработки данных до управления процессами ОС.

Примерами консольных приложений могут быть утилиты командной строки в ОС Windows или Unix (например, ping, ipconfig, ls), программы для работы с базами данных или сетевыми протоколами, а также различные инструменты для администрирования серверов.

1.2. Создание проекта консольного приложения в среде Visual Studio

Создание проекта любого консольного приложения на языке программирования C# в среде разработки Visual Studio состоит из нескольких простых шагов:

- 1) открыть Visual Studio и выбрать окно «Создать проект»;
 - 2) в открывшемся окне выбрать опцию «Консольное приложение»;
 - 3) ввести имя проекта и нажать «Создать»;
 - 4) после создания проекта откроется файл Program.cs, в котором уже содержится шаблон кода для консольного приложения (например, метод Main);
 - 5) внести необходимые изменения в код, чтобы приложение выполняло соответствующие функции;
 - 6) сохранить изменения и запустить приложение, нажав клавишу F5 или выбрав Debug > Start Debugging из меню Visual Studio.
- Ниже в качестве примера представлен код приложения, функцией которого является вывод в консоль приветственного сообщения «Привет, мир!» (листинг 1.1).

Листинг 1.1. Консольное приложение для вывода в консоль строки «Привет, мир!»

```
using System;

namespace MyConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Привет, мир!");
            Console.ReadLine(); // ожидание ввода
        }
    }
}
```

Организация ввода/вывода в консольном режиме Visual Studio на языке C# — это набор классов и методов, предназначенных для работы с консольным вводом и выводом данных в приложениях, написанных на языке C#.

Класс Console в C# предоставляет методы для ввода и вывода данных в консольном режиме, в частности Console.WriteLine, Console.Write, Console.ReadLine и т. д. Эти методы позволяют вывести текстовые сообщения на экран, читать ввод пользователя и выполнять другие операции ввода/вывода.

Кроме того, в языке C# можно использовать форматированный вывод с помощью метода Console.WriteLine, который позволяет форматировать выходные данные, например использовать специальные символы для выравнивания или форматирования числовых значений.

Для записи (вывода) информации в консоль существуют следующие методы класса Console: Write (*выводимая строка в кавычках или переменная строкового типа*), WriteLine (*выводимая строка в кавычках или переменная строкового типа*), для чтения (ввода) из консоли — метод ReadLine(), который возвращает введенную строку. Метод ReadLine всегда возвращает тип string, поэтому в случае необходимости преобразования в численный тип следует использовать преобразование данных. Параметром методов Write, WriteLine тоже должна быть переменная типа string. Однако при выводе информации в консоль можно обойтись и без преобразования. Разница между методами Write и WriteLine состоит в том, что после вывода строки WriteLine переводит курсор на следующую строку консоли, Write оставляет курсор в текущей строке после выведенных данных.

Для преобразования типов данных предназначены методы класса Convert. Метод.ToInt32 переводит в целочисленный тип int, метод.ToDouble переводит в вещественный тип double, метод.ToString предназначен для перевода в строковый тип string. Все доступные методы класса Convert можно узнать благодаря подсказкам среды Visual Studio: для этого нужно набрать имя класса, поставить точку, и среда разработки покажет полный список его методов.

Содержание

Предисловие.....	3
ДОМАШНЕЕ ЗАДАНИЕ № 1	
Консольный калькулятор арифметических выражений	5
Теоретическая часть	5
Практическая часть	16
Вопросы и задания	16
ДОМАШНЕЕ ЗАДАНИЕ № 2	
Часть 1. HTTP-сервер, отдающий статичную html-страницу	18
Теоретическая часть	18
Практическая часть	28
Вопросы и задания	29
ДОМАШНЕЕ ЗАДАНИЕ № 2	
Часть 2. Backend с форматом JSON ответа на запрос.....	30
Теоретическая часть.....	30
Практическая часть	35
Вопросы и задания	36