Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 14

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Классы и объекты»

Выполнил: ст. гр. ТУУ-111

Михалин. А. В. Вариант №2

1.06.2025

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

(дата приёмки)

Москва – 2025 г.

**1. Цель работы**

В среде *Microsoft Visual Studio* на языке *Visual C#* в консольном режиме составить программное обеспечение для решения типовых задач программирования по тематике «Классы и объекты» («*Classes and Objects*»), заданных по варианту.

**2. Формулировка задачи**

# Создайте класс: «Интернет-сайт».

Возможное имя класса: «*clsSite*».

**Поля (*fields*):**

– уникальный номер сайта в списке (тип: *short*);

– протокол доступа к сайту (тип: *string*);

– доменная зона сайта (тип: *string*);

– хостинг / владелец сайта (тип: *string*);

– верхние уровни домена сайта (тип: *string[]*);

– *IP*-адрес сайта (тип: *long*);

– название заглавной страницы сайта (тип: *string*);

– название текущей просматриваемой страницы сайта (тип: *string*);

– дата посещения сайта (тип: *long*);

– признак защищённости протокола (тип: *bool*).

**Конструктор (*instance constructor*):**

1. Пять входных параметров\*:

– адресная строка сайта,

– строка с заголовком сайта,

– строка с названием текущей страницы сайта,

– строка *IP*-адреса (в формате *IPv4*),

– строка с датой посещения сайта (в формате дд.мм.гггг).

\* – идентификатор сайта не хранится в файле, он назначается по порядковому номеру строки, содержащей, собственно, сайт и его параметры.

**Методы (*methods*):**

1. Формирование адресной строки сайта для вставки в браузер по параметрам сайта: (имя: *createSiteString*; тип: *string*).

2. Срез верхних уровней домена из адресной строки сайта, сформированной для браузера (имя: *getUpperDomainLevels*; тип: *string[]*).

3. Формирование библиографической ссылки\*\* на сайт (имя: *formatBibliography*; тип: *-*).

4. Изменение даты посещения сайта по входному строковому представлению (имя: *changeVisitDateByString*; тип: *long*).

5. Определение защищённости\*\*\* протокола (имя: *detectSecurity*; тип: *bool*).

\*\*\* – последняя буква «*s*» в названии протокола является признаком защищённости протокола, в частности *http*.

**Хранение элементов (*storage*):**

– адреса сайтов и их параметры записаны в текстовый файл; файл дополняется и обновляется только вручную через «Блокнот»; изменения, выполняемые в процессе работы программы, не сказываются на содержимом файла;

– массивы и списки для хранения не предусмотрены.

**Вывод (*ToString()*):**

Выводить информацию о сайте в формате:

«Яндекс [[*https://ya.ru*](https://ya.ru/)] – *IP*: 5.255.255.242»

«Ресторан *MANSARD* [[*http://mansard-moscow.ru*](http://mansard-moscow.ru/)] – *IP*: 185.165.123.36»

\*\* – библиографическая ссылка на сайт: «{название сайта} [Электронный ресурс] : {название текущей страницы сайта}. *URL*: {адресная строка сайта} (дата обращения: {дата в формате дд.мм.гггг})»

Параметрические скобки {} – не печатаемая информация.

Пример: «*Lenta.ru* – Новости России и мира сегодня [Электронный ресурс] : Эффективные кадры. Московский метрополитен увеличивает набор водителей трамвая в два раза. *URL*: *https:/​/​lenta.ru/​articles/​2022/​03/​18/​moskkadry/* (дата обращения: 23.03.2025)».

**3. Блок-схема алгоритма**



Рисунок 3.1 – Блок-схема алгоритма



Рисунок 3.2 – Блок-схема алгоритма



Рисунок 3.3 – Блок-схема алгоритма



Рисунок 3.4 – Блок-схема алгоритма



Рисунок 3.5 – Блок-схема алгоритма



Рисунок 3.6 – Блок-схема алгоритма



Рисунок 3.7 – Блок-схема алгоритма



Рисунок 3.8 – Блок-схема алгоритма



Рисунок 3.9 – Блок-схема алгоритма

**4. Подбор тестовых примеров**

Ожидаемый вывод программы :

ToString()

Ресторан MANSARD [https://mansard-moscow/] - IP: 185.165.123.36

createSiteString()

https://mansard-moscow.ru

getUpperDomainLevels()

https://mansard-moscow

ru

formatBibliography()

Ресторан MANSARD [Электронный ресурс] : Главная. URL: https://mansard-moscow/Главная/ (дата обращения: 23.05.2025)

detectSecurity()

Is secure protocol: True

-------------------------------------------------------------

Нажмите любую клавишу чтобы продолжить

**5. Листинг (код) программы**

using System;

using System.Collections.Generic;

public class clsSite

{

private static short counter = 0; // Статический счётчик для уникального ID

private short id; // Уникальный идентификатор сайта

private string[] protocol; // Протокол (http, https и т.д.)

private string domainZone; // Доменная зона (например, "com", "ru")

private string hosting; // Хостинг (предпоследний уровень домена)

private string[] upperDomains; // Все уровни домена (массив)

private string ip; // IP-адрес в виде числа

private string mainPageTitle; // Заголовок главной страницы

private string currentPageTitle; // Заголовок текущей страницы

private object visitDate; // Дата посещения

private bool isSecure; // Флаг безопасности (https = true)

public clsSite(string address, string mainTitle, string currentTitle, string ipString, long dateString)

{

this.id = ++counter; // Присваиваем уникальный ID объекту

string uri = address; // Парсим URL

protocol = uri.Split(':'); // Сохраняем протокол

isSecure = detectSecurity(); // Определяем безопасность (https ли)

string host = uri;

upperDomains = host.Split('.');// Разбиваем домен на части

domainZone = upperDomains[upperDomains.Length - 1]; // Получаем доменную зону

hosting = upperDomains.Length > 1 ? upperDomains[upperDomains.Length - 2] : upperDomains[0]; // Получаем "хостинг" — предпоследний уровень домена

this.mainPageTitle = mainTitle;

this.currentPageTitle = currentTitle;

ip = convertIpToLong1(ipString); // Конвертируем IP из строки в число

this.visitDate =(ChangeVisitDateByString(dateString)); // Конвертируем дату посещения из строки в Unix-время

}

public object ChangeVisitDateByString(long dateString)

{

// Конвертация даты в формат "дд.мм.гггг"

long dt = (dateString);

long ts1 = ((dt) / 1000000);

long ts2 = (((dt) / 10000) % 100);

long ts3 = ((dt) % 10000);

if ((int)(ts1 / 10) + (int)(ts2 / 10) == 0)

return "0" + ts1 + "." + "0" + ts2 + "." + ts3;

else if ((int)(ts2 / 10) == 0)

return ts1 + "." + "0" + ts2 + "." + ts3;

else if ((int)(ts1 / 10) == 0)

return "0" + ts1 + "." + ts2 + "." + ts3;

else

return ts1 + "." + ts2 + "." + ts3;

}

public string createSiteString()

{

// Формируем URL в виде "протокол://домен"

return hosting + "." + domainZone;

}

public string createSiteString1()

{

// Формируем URL в виде "протокол://домен"

return hosting + "/";

}

public string[] getUpperDomainLevels()

{

// Возвращаем массив уровней домена

return upperDomains;

}

public void formatBibliography()

{

//Вывод формата библиографической ссылки с датой посещения

object dateStr = (visitDate);

Console.WriteLine(mainPageTitle + " [Электронный ресурс] : " + currentPageTitle +

". URL: " + createSiteString1() + currentPageTitle + "/"+

" (дата обращения: " + dateStr + ")");

}

public bool detectSecurity()

{

// Если протокол заканчивается на 's' (https), возвращаем true

return protocol[0].ToLower().EndsWith("s");

}

private string convertIpToLong1(string ipString)

{

// вызов Ip

return ipString;

}

public override string ToString()

{

// Возвращает строковое представление объекта: заголовок и IP-адрес

string ipStr = (ip);

return mainPageTitle + " [" + createSiteString1() + "] – IP: " + ipStr;

}

}

class Program

{

static void Main()

{// Установка цвета фона консоли белым

Console.BackgroundColor = ConsoleColor.White;

// Установка цвета текста консоли черным

Console.ForegroundColor = ConsoleColor.Black;

// Применяем изменения (очищая окно, чтобы фон изменился у всего окна)

Console.Clear();

// Создаём объект clsSite с информацией о сайте

clsSite site = new clsSite(

"https://mansard-moscow.ru", // URL сайта

"Ресторан MANSARD", // Заголовок главной страницы

"Главная", // Заголовок текущей страницы

"185.165.123.36", // IP-адрес сайта

23052025 // Дата посещения сайта в формате дд.мм.гггг

);

// Выводим информацию, вызвав метод ToString() класса clsSite

Console.WriteLine("ToString()");

Console.WriteLine(site.ToString());

// Выводим полную строку формирования URL

Console.WriteLine("createSiteString()");

Console.WriteLine(site.createSiteString());

// Получаем и выводим уровни домена (части доменного имени)

Console.WriteLine("getUpperDomainLevels()");

string[] levels = site.getUpperDomainLevels();

foreach (string level in levels)

Console.WriteLine(level);

// Форматируем и выводим библиографическую ссылку с информацией о сайте

Console.WriteLine("formatBibliography()");

site.formatBibliography();

// Проверяем, использует ли сайт защищённый протокол (HTTPS)

Console.WriteLine("detectSecurity()");

Console.WriteLine("Is secure protocol: " + site.detectSecurity());

// Ожидаем нажатия клавиши, чтобы программа не закрылась сразу

Console.WriteLine("-------------------------------------------------------------");

Console.WriteLine("Нажмите любую клавишу чтобы продолжить");

Console.ReadKey();

}

}

**6. Расчёт тестовых примеров на ПК**

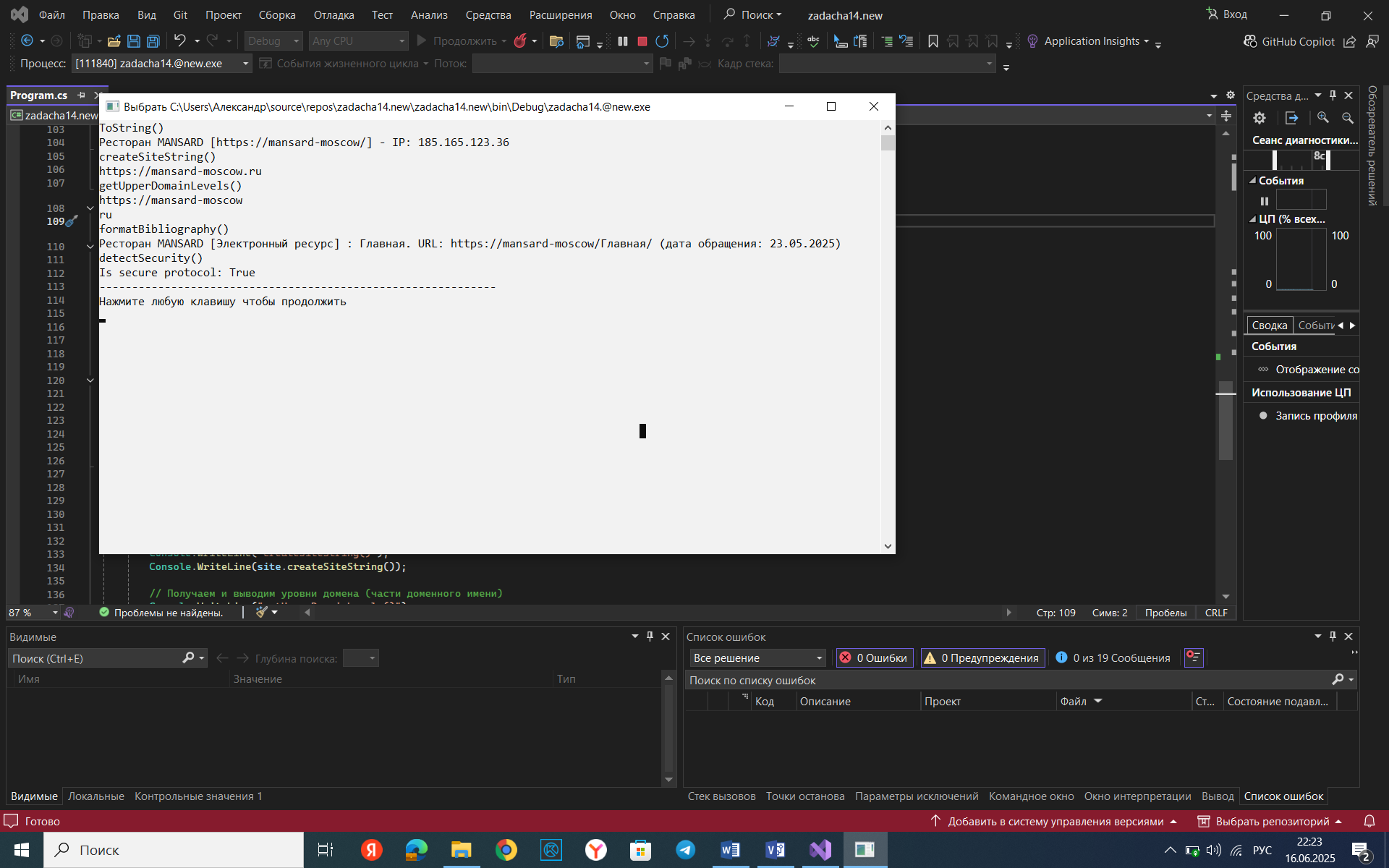
****

Рисунок 6.1 – Результат работы всей программы

**7. Вывод по работе**

В результате выполнения данной работы было составлено программное обеспечения для решения задания по индивидуальному варианту. Были получены навыки работы с классами и объектами на языке *Visual C#*.