***Работа с буфером обмена: назначение буфера обмена, форматы данных, функции для работы с буфером обмена.***

***Дано натуральное число n и последовательность из 7 чисел. Найти количество чисел, являющихся степенями пятерки. В программе разработать функцию, позволяющую распознавать степень пятерки.***

**Назначение буфера обмена** — промежуточное хранилище, в которое помещаются определённые данные, когда пользователь операционной системы копирует (Ctrl+C) или вырезает (Ctrl+X) какую-либо информацию (текст из текстового редактора, группа файлов из каталога и т.п.). Когда пользователь в дальнейшем вставляет скопированные/вырезанные данные (Ctrl+V), копия этих данных извлекается из буфера обмена и помещается в указанное место.

Форматы данных буфера обмена — это строки, идентифицирующие формат для получения связанных данных приложением, использующим этот формат. Класс DataFormats предоставляет для использования предопределённые имена форматов. Также можно создавать собственные имена форматов или задать тип объекта в качестве формата.

**Функции для работы с буфером обмена в C#:**

SetText(string) — помещает текст (переданный в качестве аргумента) в буфер обмена;

GetText() — возвращает текст из буфера обмена (помещённый туда до этого);

Clear() — удаляет содержимое буфера обмена;

ContainsText() — если в буфере обмена содержится текст — метод вернёт true (в противном случае — false).

**Назначение буфера обмена**

Обмен данными между приложениями: Буфер обмена (clipboard) — это механизм, предоставляемый операционной системой, который позволяет пользователям обмениваться данными между различными приложениями.

Временное хранилище: Он служит как временное хранилище для копируемых данных, откуда их можно вставить в другое место (обычно в другое приложение).

Копирование и вставка: Буфер обмена является основой операций “копировать” (Ctrl+C, Cmd+C) и “вставить” (Ctrl+V, Cmd+V), а также “вырезать” (Ctrl+X, Cmd+X)

Различные типы данных: Буфер обмена может хранить различные типы данных, от простого текста до изображений, HTML, файлов и других структурированных данных.

Глобальный ресурс: Буфер обмена является глобальным ресурсом операционной системы, что позволяет приложениям взаимодействовать друг с другом, даже если они не связаны напрямую.

**Форматы данных в буфере обмена**

Представление данных: Буфер обмена не просто хранит “сырые” данные, а хранит их в определенном формате. Это позволяет принимающему приложению интерпретировать данные правильно.

Стандартные форматы: Существуют стандартные форматы, которые широко поддерживаются большинством приложений:

CF\_TEXT (или System.Windows.Forms.DataFormats.Text в .NET): Обычный текст.

CF\_UNICODETEXT (или System.Windows.Forms.DataFormats.UnicodeText в .NET): Текст в Unicode (UTF-16).

CF\_RTF (или System.Windows.Forms.DataFormats.Rtf в .NET): Rich Text Format (RTF) для текста с форматированием.

CF\_BITMAP (или System.Windows.Forms.DataFormats.Bitmap в .NET): Растровое изображение (обычно в формате DIB).

CF\_DIB: Device Independent Bitmap (формат растрового изображения, не привязанный к конкретному устройству).

CF\_HTML: HTML контент.

Пользовательские форматы: Приложения могут регистрировать свои собственные форматы для обмена данными, специфичными для данного приложения.

Идентификация формата: Каждому формату данных соответствует числовой идентификатор или строка (например, MIME-тип).

**Функции для работы с буфером обмена в C#**

В C# для работы с буфером обмена обычно используется класс System.Windows.Forms.Clipboard (в WinForms) или System.Windows.Clipboard (в WPF) из пространства имен System.Windows.

**Получение данных из буфера обмена:**

Clipboard.GetDataObject(): Возвращает объект типа IDataObject, содержащий все данные, хранящиеся в буфере обмена.

Clipboard.GetText(): Возвращает текст из буфера обмена, если он доступен в формате CF\_TEXT или CF\_UNICODETEXT.

Clipboard.GetImage(): Возвращает изображение из буфера обмена, если оно доступно в формате CF\_BITMAP.

Clipboard.GetData(string format) или Clipboard.GetData(DataFormats format): Возвращает данные из буфера обмена в указанном формате.

**Установка данных в буфер обмена:**

Clipboard.SetDataObject(object data): Устанавливает данные в буфер обмена. Можно передать объект, реализующий интерфейс IDataObject, или простой объект, который автоматически будет сконвертирован в нужные форматы.

Clipboard.SetText(string text): Устанавливает текст в буфер обмена.

Clipboard.SetImage(Image image): Устанавливает изображение в буфер обмена.

Clipboard.SetData(string format, object data) или Clipboard.SetData(DataFormats format, object data): Устанавливает данные в буфер обмена в указанном формате.

**Проверка наличия данных:**

Clipboard.ContainsData(string format) или Clipboard.ContainsData(DataFormats format): Возвращает true, если буфер обмена содержит данные в указанном формате.

Clipboard.ContainsText(): Проверяет наличие текстовых данных.

Clipboard.ContainsImage(): Проверяет наличие изображения.

**Очистка буфера обмена:**

Clipboard.Clear(): Удаляет все данные из буфера обмена.

**Пример (WinForms):**

using System;

using System.Windows.Forms;

using System.Drawing;

public class ClipboardExample

{

public static void Main(string[] args)

{

// Копирование текста в буфер обмена

Clipboard.SetText("Hello from C#!");

// Проверка наличия текста

if (Clipboard.ContainsText())

{

string text = Clipboard.GetText();

Console.WriteLine("Текст из буфера обмена: " + text);

}

// Копирование изображения в буфер обмена

Bitmap bitmap = new Bitmap("image.png"); // Предполагается, что файл image.png существует

Clipboard.SetImage(bitmap);

// Проверка наличия изображения

if (Clipboard.ContainsImage())

{

Image image = Clipboard.GetImage();

Console.WriteLine("Изображение в буфере обмена: " + (image != null ? "Есть" : "Нет"));

// Здесь можно, например, сохранить изображение из буфера в файл.

}

// Копирование RTF текста в буфер обмена

Clipboard.SetData(DataFormats.Rtf, "{<[\\rtf1\\ansi This is a test with \\b bold \\b0](file:///\\rtf1\ansi%20This%20is%20a%20test%20with%20\b%20bold%20\b0)> text.}");

if (Clipboard.ContainsData(DataFormats.Rtf))

{

string rtfText = Clipboard.GetData(DataFormats.Rtf) as string;

Console.WriteLine("RTF из буфера: " + rtfText);

}

Console.ReadKey();

}

}