using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

class Program

{

static void Main()

{

bool continueCreating = true;

while (continueCreating)

{

Console.Clear();

Console.WriteLine(" СИСТЕМА СОЗДАНИЯ ДОКУМЕНТОВ ");

Console.WriteLine("Какой документ хотите создать?");

Console.WriteLine("1 - Текстовый");

Console.WriteLine("2 - Изображение");

Console.WriteLine("3 - Таблицу");

Console.WriteLine("0 - Выход");

Console.Write("Ваш выбор: ");

string choice = Console.ReadLine();

switch (choice)

{

case "1":

CreateTextDocument();

break;

case "2":

CreateImageDocument();

break;

case "3":

CreateSpreadsheetDocument();

break;

case "0":

continueCreating = false;

Console.WriteLine("Работа завершена. До свидания!");

break;

default:

Console.WriteLine("Неверный выбор");

break;

}

if (continueCreating && choice != "0")

{

Console.WriteLine("\nНажмите любую клавишу для продолжения...");

Console.ReadKey();

}

}

}

static void CreateTextDocument()

{

TextDocument doc1 = new TextDocument();

TextDocument doc2 = new TextDocument();

doc1.InitializeTextSettings();

Console.Write("Введите название: ");

doc1.Title = Console.ReadLine();

Console.Write("Введите автора: ");

doc1.Author = Console.ReadLine();

Console.Write("Введите содержание: ");

doc1.Content = Console.ReadLine();

Console.Write("Введите размер шрифта: ");

doc1.FontSize = int.Parse(Console.ReadLine());

Console.WriteLine();

doc1.DisplayDocument();

Console.WriteLine();

doc2.DisplayDocument();

}

static void CreateImageDocument()

{

ImageDocument doc = new ImageDocument

{

Title = "Новое изображение",

Author = "Фотограф"

};

Console.Write("Введите название: ");

doc.Title = Console.ReadLine();

Console.Write("Введите автора: ");

doc.Author = Console.ReadLine();

Console.Write("Введите описание: ");

doc.Content = Console.ReadLine();

Console.WriteLine("Выберите разрешение:");

Console.WriteLine("1 - 1280x720 (HD)");

Console.WriteLine("2 - 1920x1080 (Full HD)");

Console.WriteLine("3 - 3840x2160 (4K)");

Console.WriteLine("4 - 7680x4320 (8K)");

Console.Write("Ваш выбор: ");

string resolutionChoice = Console.ReadLine();

switch (resolutionChoice)

{

case "1":

doc.Resolution = "1280x720 (HD)";

break;

case "2":

doc.Resolution = "1920x1080 (Full HD)";

break;

case "3":

doc.Resolution = "3840x2160 (4K)";

break;

case "4":

doc.Resolution = "7680x4320 (8K)";

break;

default:

doc.Resolution = "Неизвестно";

break;

}

Console.WriteLine();

doc.DisplayDocument();

}

static void CreateSpreadsheetDocument()

{

Console.Write("Введите количество строк: ");

int rows = int.Parse(Console.ReadLine());

Console.Write("Введите количество столбцов: ");

int columns = int.Parse(Console.ReadLine());

SpreadsheetDocument doc = new SpreadsheetDocument

{

Title = "Новая таблица",

RowCount = rows,

ColumnCount = columns

};

Console.Write("Введите название: ");

doc.Title = Console.ReadLine();

Console.Write("Введите автора: ");

doc.Author = Console.ReadLine();

Console.Write("Введите описание: ");

doc.Content = Console.ReadLine();

Console.WriteLine();

doc.DisplayDocument();

}

}

class Document

{

public string Title;

public string Author;

public string Content;

public DateTime CreatedDate;

public Document()

{

Title = "Без названия";

Author = "Неизвестный автор";

Content = "Пусто";

CreatedDate = DateTime.Now;

Console.WriteLine($"Создан новый документ: {CreatedDate}");

}

public Document(string title, string author, string content)

{

Title = title;

Author = author;

}

~Document()

{

Console.WriteLine($"Документ '{Title}' удален");

}

public void ResetToDefault()

{

Title = "Без названия";

Author = "Неизвестный автор";

Content = "Пусто";

Console.WriteLine("Документ сброшен к значениям по умолчанию");

}

public virtual void DisplayDocument()

{

Console.WriteLine($"Документ: {Title}");

Console.WriteLine($"Автор: {Author}");

Console.WriteLine($"Содержание: {Content}");

}

}

class TextDocument : Document

{

public int FontSize;

public void InitializeTextSettings()

{

if (FontSize <= 0) FontSize = 12;

if (string.IsNullOrEmpty(Content)) Content = "Текстовый документ";

Console.WriteLine("Текстовые настройки инициализированы");

}

public TextDocument() : base()

{

FontSize = 14;

Console.WriteLine("Конструктор TextDocument вызван");

}

public TextDocument(string title, string author, string content, int fontSize)

: base(title, author, content)

{

FontSize = fontSize;

Console.WriteLine($"Конструктор TextDocument с параметрами вызван: {title}");

}

public void InitializeDefault()

{

if (string.IsNullOrEmpty(Title))

Title = "Текстовый документ";

if (FontSize <= 0)

FontSize = 12;

Console.WriteLine("TextDocument инициализирован значениями по умолчанию");

}

public override void DisplayDocument()

{

InitializeDefault();

Console.WriteLine("ТЕКСТОВЫЙ ДОКУМЕНТ");

base.DisplayDocument();

Console.WriteLine($"Размер шрифта: {FontSize}");

}

}

class ImageDocument : Document

{

public string Resolution;

public ImageDocument() : base()

{

Resolution = "1920x1080 (Full HD)";

}

public void ValidateResolution()

{

if (string.IsNullOrEmpty(Resolution))

{

Resolution = "1920x1080 (Full HD)";

Console.WriteLine("Разрешение установлено по умолчанию");

}

else

{

Console.WriteLine($"Разрешение проверено: {Resolution}");

}

}

public override void DisplayDocument()

{

ValidateResolution();

Console.WriteLine("ИЗОБРАЖЕНИЕ");

base.DisplayDocument();

Console.WriteLine($"Разрешение: {Resolution}");

}

}

class SpreadsheetDocument : Document

{

public int RowCount;

public int ColumnCount;

public SpreadsheetDocument() : base()

{

RowCount = 1;

ColumnCount = 1;

}

public void ValidateDimensions()

{

if (RowCount <= 0)

{

RowCount = 1;

Console.WriteLine("Количество строк исправлено на 1");

}

if (ColumnCount <= 0)

{

ColumnCount = 1;

Console.WriteLine("Количество столбцов исправлено на 1");

}

Console.WriteLine($"Размеры таблицы проверены: {RowCount}x{ColumnCount}");

}

public override void DisplayDocument()

{

ValidateDimensions();

Console.WriteLine("ТАБЛИЦА");

base.DisplayDocument();

Console.WriteLine($"Размер таблицы: {RowCount}x{ColumnCount}");

}

}

/\*

ЗАДАНИЕ ОТ САГЫНАЙ МАРАТОВНЫ НА ПАРУ

Дополнить код любыми 2 конструкторами

Дополнить одним деструктором

Доплнить код 2 иникиализатора

ЗАДАНИЕ ДЛЯ ПОЛНОГО ПОНИМАНИЕ, УСВОЕНИЯ И НАРАБОТКИ

Додумать,что то не стандартное,дополнительное

Работа с файлами (сохранение, чтение, изменение и т.д.)

Вставить в код, задание от Сагынай Маратовны в правильном порядке

Придумать более сложное решение

Придкмать доплнительное условия

!!!НЕ УМИРЕТЬ!!! - ЖЕЛАТЕЛЬНО

\*/