

КОМПЛЕКС СРЕДСТВ ДЛЯ БЫСТРОГО ПОСТРОЕНИЯ ФАЙЛОВ С ПРОИЗВОЛЬНОЙ СТРУКТУРОЙ

Мацкевич А.С.,

*учащийся 3 курса Оршанского колледжа ВГУ имени П.М. Машерова,
г. Орша, Республика Беларусь*

Научный руководитель – Романцов Д.Ю., магистр техн. наук, преподаватель

Ключевые слова. C#, шифрование, файлы, структуры, данные.

Keywords. C#, encryption, files, structures, data.

В настоящее время существует множество различных форматов файлов, каждый из которых предназначен для хранения определенного типа данных. Однако, часто появляется необходимость записать данные, не подходящие ни под один из существующих форматов, а также работать с произвольными структурами данных. Ещё может возникнуть задача для построения сложного файла, например, содержащего какую-то модификацию для программы. Это могут быть как какие-то переменные, так и различные сторонние файлы, скрипты, текстуры. Реализация в коде при помощи пространства имён System.IO выглядит громоздко и сложно для восприятия сторонним программистом не знакомым с кодом программы и её особенностями.

Целью исследования является создание комплекса средств для быстрого построения файлов с произвольной структурой. Разработка библиотеки, помогающей более удобно сохранять данные, а также программного средства способного открывать файлы такого типа для визуализации содержимого и его редактирования.

Для достижения этой цели были поставлены следующие задачи:

1. Изучение существующих форматов файлов и их структур
2. Разработка алгоритма создания структуры файла
3. Разработка библиотеки
4. Разработка программы визуального представления данных

Материал и методы. Разработка программного средства для визуализации производилась в Visual Studio 2022 и языком программирования C#. Был выбран тип приложения Windows Forms.

Для начала были изучены размеры различных типов данных (рис. 1) [3].

Тип	Размер в байтах	Диапазон значений
char	1	от -128 до 126
unsigned char	1	от 0 до 255
short	2	от -32 768 до 32767
unsigned short	2	от 0 до 65536
enum	2	от -2 147 483 648 до 2 147 483 647
long	4	от -2 147 483 648 до 2 147 483 647
unsigned long	4	от 0 до 4 294 967 295
int	4	от -2 147 483 648 до 2 147 483 647
unsigned int	4	от 0 до 4 294 967 295
float	4	от $3,4 * 10^{-38}$ до $3,4 * 10^{38}$
double	8	от $1,7 * 10^{-308}$ до $3,4 * 10^{308}$
long double	10	от $3,4 * 10^{-4932}$ до $1,1 * 10^{4932}$
bool	1	true или false

Рисунок 1 – Типы данных и их размер в байтах

После этого был разработан алгоритм для записи данных со структурой. Она представляет собой отдельный файл, который содержит в себе основную информацию о размерах и количествах. Не имея файла структуры, невозможно открыть файл и извлечь содержимое (рис. 2).

В последующем была разработана библиотека, имеющая возможности по созданию таких файлов со структурами. Она была разработана с использованием ООП и его основных концепций: инкапсуляции, наследования, полиморфизма (рис. 3) [1; 2].

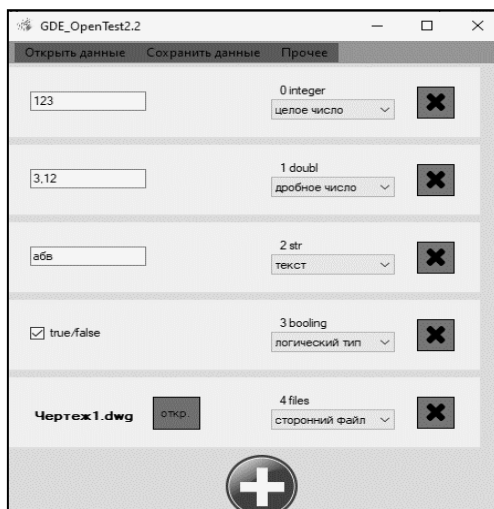


Рисунок 2 – Приложение для визуализации данных

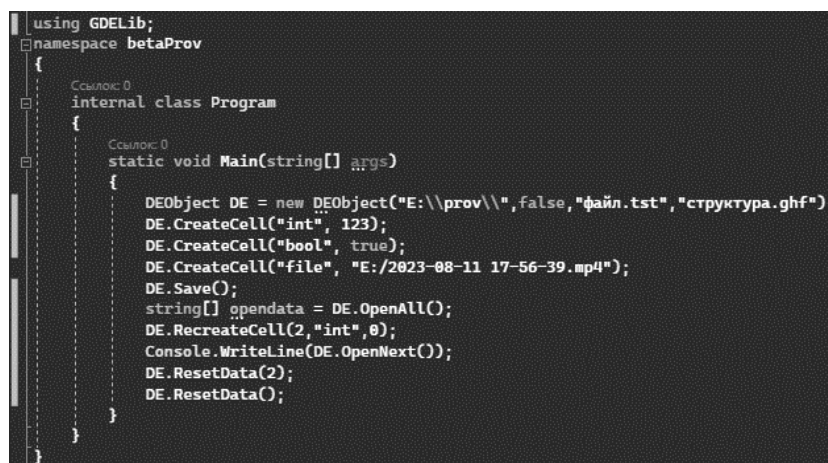


Рисунок 3 – Использование основных возможностей библиотеки

Результаты и их обсуждение. По итогу разработки было получено программное средство для визуализации данных и библиотека для записи их в произвольных формат на усмотрение программиста. Благодаря использованию ООП и инкапсуляции, разработчик не может на прямую обращаться и изменять данные в объектах библиотеки. Это повышает надёжность работы и понятность разработки приложений с записью данных.

Из-за отдельного сохранения данных, а именно файл со структурой и файл с данными, которые представляют из себя бинарные файлы. Это повышает защищённость, потому что если попытаться открыть файлы в блокноте или других текстовых редакторах, то невозможно понять, что в нём содержится. Так же отдельное сохранение способствует защите данных, потому что их невозможно извлечь без файла структуры, она является своеобразным ключом.

Так же в библиотеке была реализована возможность сохранять все данные и структуру в 1 файл. Этот тип сохранения может использоваться там, где не нужно обеспечивать надёжность в виде отдельной структуры [4], например, различные модификации для приложений. Так же она идеально подходит в случаях, где непостоянное количество данных – различные редакторы, меню товаров, справочники, игры и т.д.

Было проведено множество тестов алгоритма сохранения и представления файла. Также с использованием библиотеки была разработана игра, пошаговая стратегия с случайной генерацией карты (рис. 4).

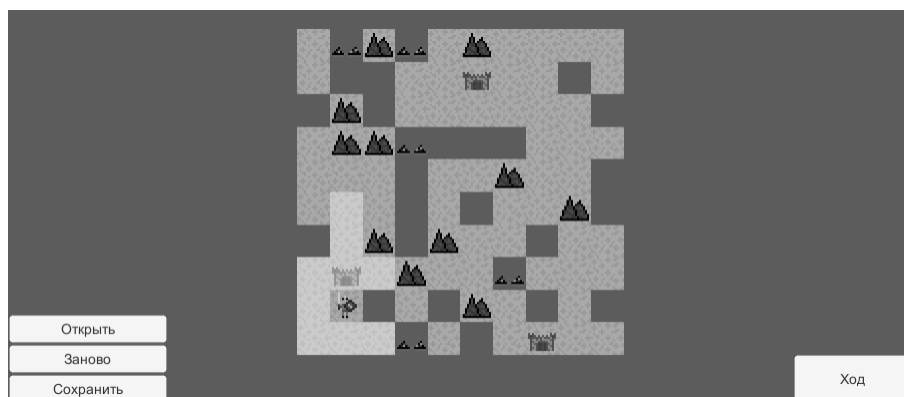


Рисунок 4 – Пошаговая стратегия

В качестве недостатка можно выявить проблемы с асинхронностью процессов, а также хотелось бы реализовать сохранение массивов и квадратных массивов (матриц), как отдельный тип данных. Это повысило бы простоту работы с библиотекой, чтения и использования данных в последующем. В качестве безопасности так же хотелось бы сделать что-то на подобие пароля, чтобы сторонние разработчики могли защитить свои данные даже если структура и основной файл сохраняются не раздельно.

Заключение. По итогу исследования были изучены виды представления данных и реализация для их сохранения при помощи разработанной библиотеки и программного средства для графического представления данных. Были выявлены недостатки и предложения для дальнейшей разработки.

1. Глухова, Л.А. Технология разработки программного обеспечения: учеб. пособие для студентов специальности 1-40 01 01 «Программное обеспечение информационных технологий» / Л. А. Глухова. – Минск : БГУИР, 2007.
2. Либерти, Д. Программирование на C#. – Пер. с англ. / Д. Либерти. – СПб : Символ-Плюс, 2003. – 688 с., ил.
3. Эндриесс, Д. Практический анализ двоичных файлов – пер. с англ. / А. А. Слинкина. – М. : ДМК Пресс, 2021. – 460 с.
4. ГОСТ 28147–89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. – М. : Изд-во стандартов, 1989.

МЕТОДИКА РАЗРАБОТКИ ИНТЕГРИРОВАННЫХ САПР И ИХ ПРИМЕНЕНИЕ В ПРОИЗВОДСТВЕ

Петров Д.А.,

студент 4 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь

Научные руководители – Буевич А.Э., канд. техн. наук, доцент;

Буевич Т.В., канд. техн. наук, доцент

Ключевые слова. Интегрированная система, автоматизированное проектирование, макрос, управляющая программа.

Keywords. Integrated system, computer-aided design, macro, control program.

Интегрированная САПР – это программный продукт, обеспечивающий работу нескольких разнородных систем с единым интерфейсом и возможностью обмена данными с внешними приложениями. Нередко системы автоматизированного проектирования и управления предприятия не позволяют решить возникающие производственные задачи. Один из способов доработки – интегрирование в действующие системы модулей, расширяющих их возможности, и формирование интегрированной системы управления.