

механизмов безопасности для реальной системы зависит от её индивидуальных особенностей.

### Заключение

Мультимедийный парк – это не просто место развлечений, а высокотехнологичный критический объект, где цифровая и физическая безопасность неразделимы. Традиционные подходы к информационной безопасности здесь недостаточны. Требуется специализированная стратегия, учитывающая гибридную природу инфраструктуры, высочайшие требования к доступности и целостности, а также повышенные риски, связанные с обработкой биометрических данных. Упреждающие инвестиции в построение адаптивной, многослойной системы защиты являются не статьей расходов, а ключевым условием устойчивого бизнеса, сохранения репутации и, что самое важное, безопасности тысяч посетителей, доверяющих парку свои данные и досуг.

### Список литературы

1. Вострецова Е. В. Основы информационной безопасности: учебное пособие, 2019. 208 с.
2. Буданов Д. Нюансы организации информационной безопасности в музеях // Системы безопасности. 2025. № 2.
3. Богданов А. В., Малыгин И. Г., Синешук Ю. И. Неформальная модель нарушителя безопасности объектов культуры // Вестник Санкт-Петербургского университета ГПС МЧС России. 2013. № 3. С. 109–113. URL: [vestnik.igps.ru](http://vestnik.igps.ru) (дата обращения: 18.10.2025).
4. Корчагин С. И., Павлов В. Г., Бутов А. Н., Ткаченко Д. Г. Подходы к созданию систем обеспечения безопасности особо важных объектов // Системы безопасности. 2010. № 4.
5. Семкин С. Н., Беляков Э. В., Гребенев С. В., Козачок В. И. Основы организационного обеспечения информационной безопасности объектов информатизации: учеб. пособие. М.: Гелиос АРВ, 2005.
6. Климов С. М. Методы и модели противодействия компьютерным атакам. Люберцы: КАТАЛИСТ, 2008.
7. Пресс-конференция о работе мультимедийных исторических парков «Россия – моя история» [Электронный ресурс]. URL: <http://pressmia.ru/pressclub/20171213/951788008.html> (дата обращения: 18.10.2025).
8. Технологии интеллектуального музея нуждаются в этих 7 решениях Интернета вещей // Умный турист. 2022, 27 июня.

### ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПРОГНОЗИРОВАНИЯ ОБЪЁМА ВЫПУСКА ПРОДУКЦИИ ПРЕДПРИЯТИЯ

Шипилов Д. В., Мозговенко А. А.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,  
e-mail: ya@mozgovenko.ru*

Цель исследования – разработать программное обеспечение на C#, обеспечивающее достоверное прогнозирование объёма выпуска продукции с учётом исторических данных, текущих заказов и производственных ограничений.

Задачи исследования:

1. Проанализировать существующие методы прогнозирования производственных показателей и выбрать оптимальные для реализации.
2. Определить требования к функциональности и интерфейсу ПО с учётом потребностей производственных менеджеров.
3. Разработать архитектуру приложения с модульной структурой (ввод данных, расчёт, визуализация, экспорт).
4. Реализовать алгоритмы обработки временных рядов и статистического анализа на C#.
5. Обеспечить интеграцию с типовыми ERP системами через API или CSV импорт/экспорт.
6. Протестировать ПО на реальных производственных данных и оценить точность прогнозов.

### Материалы и методы исследования

Современные исследования в области прогнозирования производственных объёмов фокусируются на:

- Машинном обучении. Публикации 2023–2025 гг. демонстрируют эффективность нейронных сетей (LSTM, GRU) для анализа временных рядов производства. Однако их внедрение требует больших объёмов данных и вычислительных ресурсов.
  - Классических статистических методах. ARIMA, экспоненциальное сглаживание и регрессионный анализ остаются актуальными для предприятий с ограниченной историей данных.
  - Интеграции с IoT. Исследования подчёркивают важность учёта данных с датчиков оборудования для корректировки прогнозов в реальном времени.
  - Облачных решениях. Появление SaaS-платформ для производственного планирования (например, Oracle SCM, SAP IBP) задаёт стандарты юзабилити и масштабируемости.
  - Визуализации данных. Современные работы акцентируют необходимость интерактивных дашбордов с KPI и сценарным анализом.
- Основная цель – создать гибкое, масштабируемое ПО на C#, сочетающее проверенные статистические методы и элементы машинного обучения для прогнозирования объёмов производства.

### Результаты исследования и их обсуждение

Для разработки данного программного обеспечения был использован язык программирования C#. Были задействованы следующие библиотеки (рис. 1):

System;  
System.Collections.Generic;  
System.ComponentModel;  
System.Data;  
System.Drawing;

```
System.Linq;
System.Text;
System.Threading.Tasks;
System.Windows.Forms;
System.IO;
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
```

Рис. 1. Библиотеки, которые были использованы при программировании

При загрузке основной формы программы осуществляется загрузка данных по умолчанию, использующая процедуры `ReadDataFromCSVFile` и `ReadWCFromCSVFile` (рис. 2).

Если файл `WC.CSV` отсутствует или пуст, то есть таблица коэффициентов оставшаяся не заполненная, то запускается процедура `CalculateWheightcoefficients`, которая заполняет таблицу сгенерированными коэффициентами исходя из начальных значений входных данных (рис. 3).

Принудительно запустить процедуру можно, выбрав в меню пункт «Загрузить файл CSV». При этом перед выполнением процедуры откроется окно выбора файла.

Процедуры `ReadFromCSVFile` и `ReadWCFromCSVFiles` сначала очищают соответствующие таблицы на форме, а затем считывают текстовые файлы в массив значений и заполняют соответствующие таблицы.

Процедура `CalculateWeightCoefficients` используется для расчета весовых коэффициентов по умолчанию (рис. 4), перед расчетом таблица коэффициентов очищается в зависимости от начальных значений, берется соответствующее количество строк из таблицы входных данных, высчитывается соотношение к сумме каждого значения, и сортируется по возрастанию. Сумма коэффициентов всегда равна единице, порядок коэффициентов соответствует требуемому соотношению, например  $a1 \leq a2 \leq a3$ ;  $A1 + a2 + a3 = 1$ . Принудительный запуск процедуры заполнения коэффициентов по умолчанию выполняется при нажатии кнопки «Заполнить коэффициенты» (рис. 5).

При нажатии кнопки `Рассчитать` запускается процедура `CalculateMovingAverageMethod`. Перед расчетом производится проверка на правильность заполнения коэффициентов, а также проверка заполнения и достаточность входных данных. Далее перебирая каждую строчку таблицы данных производится расчет плановых показателей. По окончании процедуры выполняется метод `CalculateSAO`, подсчитывающий сумму среднего абсолютного отклонения.

```
string defaultDataFile = Application.StartupPath + "\\data.csv";

if (File.Exists(defaultDataFile))
{
    ReadDataFromCSVFile(defaultDataFile);
    ReadWCfromCSVFile();
    if (dataGridView2.Rows.Count == 0) CalculateWeightCoefficients();
}
}
```

Рис. 2. Код загрузки основной формы по умолчанию

The screenshot shows the 'Forecasts' application window. A menu is open over the 'Рассчитать' button, listing options: 'Скачать Данные из CSV', 'Сохранить данные в CSV', 'Сохранить расчет в CSV', and 'Выход'. Below the menu is a table with 5 columns: 'Средне (простое)', 'Разница (простое)', 'Средне (взвешенное)', and 'Разница (взвешенное)'. The table contains 5 rows of data.

	Средне (простое)	Разница (простое)	Средне (взвешенное)	Разница (взвешенное)
	0,00	0,00	0,00	0,00
	107,34	49,66	115,57	41,43
	144,00	26,75	140,79	23,54
5	01.07.2020	75	125,42	50,42
			128,18	53,18

Рис. 3. Отображение возможных функций работы с файлом

```

Ссылка 1
private void CalculateSAO()
{
    int CountFistRows = Convert.ToInt32(StartCount.Value);
    decimal differentSum = 0;
    for (int i = CountFistRows; i < dataGridView1.Rows.Count-1; i++)
    {
        if (dataGridView1.Rows[i].Cells[0].Value == null) continue;
        if (Decimal.TryParse(dataGridView1.Rows[i].Cells[6].Value.ToString(), out decimal differentValue))
            differentSum += differentValue;
    }
    decimal SAO = differentSum / (dataGridView1.Rows.Count - CountFistRows);
    textBoxSAO.Text = SAO.ToString();
}
    
```

Рис. 4. Отражение весовых коэффициентов

	Коэффициент
a1	0,214285928571429
a2	0,357143214285714
a3	0,428571857142857

Рис. 5. Код расчета САВ

Form2\_load. При выборе пункта «Сохранить данные в CSV» (рис. 6).

Выбирая в меню пункт «Сохранить расчет CSV», вызывается окно выбора файла процедурой ExportResultToCSV\_click. После успешного выбора вызываются процедуры WriteResultsToCSVFile записывающей результат расчетов в файл и WriteWCToCSVFile для записи таблицы коэффициентов. Таблицы коэффициентов создаются в каталоге программы.

При закрытии программы по умолчанию запускаются процедуры сохранения таблиц формы расчетной и коэффициентов, вызывая процедуры WriteDataToCSVFile и WriteWCToCSVFile.

### Заключение

Разработанное ПО для прогнозирования объема выпуска продукции на языке C# демонстрирует высокую практическую ценность для малого и среднего бизнеса, предоставляя доступную альтернативу дорогостоящим облачным решениям.

### Список литературы

1. Жданов С. А., Соболева М. Л., Алфимова А. С. Информационные системы: учебник. М.: Прометей, 2015. 302 с.
2. Жиганов С. Н. Анализ динамических систем: учеб. пособие. Саратов: Ай Пи Эр Медиа, 2018. 202 с.
3. Исакова А. И. Предметно-ориентированные экономические информационные системы: учеб. пособие. Томск: Томский государственный университет систем управления и радиоэлектроники, 2016. 238 с.
4. Исаченко О. В. Программное обеспечение компьютерных сетей: учеб. пособие. М.: ИНФРА-М, 2017. 117 с.
5. Ковалева В. Д. Информационные системы в экономике: учеб. пособие. Саратов: Вузовское образование, 2018. 88 с.
6. Коноплева И. А. Информационные системы и технологии управления: учебник. М.: ЮНИТИ-ДАНА, 2017. 591 с.

При нажатии кнопки показать график открывается окно с графиком Form1\_load, которое при загрузке формы читает данные из таблицы расчетов данных, строит графики. За это отвечает стандартные процедуры