

СОЗДАНИЕ УНИВЕРСАЛЬНОГО КОНВЕРТЕРА ВЕЛИЧИН И КАЛЬКУЛЯТОРА НА WINDOWSFORMS

Тазетдинова Ю.А., к.ф.-м.н., доцент,

Крохалев К.О., студент,

Минлин М.В., студент

Бирский филиал УУНиТ, г. Бирск, Россия

Аннотация. В статье рассматривается разработка универсального конвертера величин и калькулятора с использованием Windows Forms. Описываются основные этапы проектирования, включая реализацию функций конвертации различных единиц измерения (длина, масса, температура и др.) и выполнения математических вычислений. Приложение предназначено для упрощения работы с разнородными данными и повышения эффективности расчетов в повседневных и профессиональных задачах.

Ключевые слова: конвертер величин, калькулятор, Windows Forms, C#, единицы измерения, математические вычисления, пользовательский интерфейс.

Введение. Автоматизация преобразования физических величин и математических вычислений – актуальная задача для образовательных, инженерных и научных приложений. В данной статье рассматривается разработка универсальной программы на платформе Windows Forms с использованием языка C# [1-3]. Программа объединяет в себе конвертер 8 категорий физических величин (длина, масса, температура, скорость, объем, площадь, время, цифровые данные) и многофункциональный калькулятор. Особое внимание уделено реализации интуитивно понятного динамического интерфейса с поддержкой смены тем оформления, а также системе обработки ошибок ввода для повышения удобства пользователей.

Цель работы – создание многофункционального инструмента для точных вычислений и конвертации.

Основные задачи разработки включают реализацию точных алгоритмов преобразования физических величин восьми категорий (длина, масса, температура, скорость, объем, площадь, время, цифровые данные) с учетом особых случаев; создание интуитивно понятного интерфейса с динамическими элементами управления для удобного взаимодействия пользователя; обеспечение надежной системы обработки и валидации вводимых данных с выводом понятных сообщений об ошибках; интеграцию функционала калькулятора с возможностью использования результатов вычислений в конвертере; а также комплексное тестирование и оптимизацию производительности приложения для гарантии стабильной работы.

Реализация проекта. Программа состоит из двух основных моделей: конвертера величин и калькулятора. Конвертер позволяет преобразовывать значения между различными единицами измерения, используя заранее заданные коэффициенты, которые хранятся в словаре. Этот словарь хранит все поддерживаемые единицы измерения и их коэффициенты преобразования относительно базовой единицы, где каждая запись содержит: название единицы, тип величины (например, длина, масса, время) и коэффициент для перевода в базовую единицу, что позволяет унифицировать расчёты и обеспечивает точность конвертации между различными системами измерений. Стандартные единицы измерения, такие как длина (метры, футы) и масса (килограммы, фунты), конвертируются путём умножения исходного значения на заранее заданный коэффициент, который определяет соотношение между единицами. Например, для перевода метров в футы значение умножается на ~ 3.28084 , а для преобразования килограммов в фунты — на ~ 2.20462 . Обратная конвертация выполняется делением на тот же коэффициент. Для более сложных единиц (например, температуры) применяются специализированные формулы, учитывающие смещение шкал, такие как $F = C \times \frac{9}{5} + 32$. Такая система обеспечивает точность и универсальность преобразований (рис. 1). Данное приложение обеспечивает быстрый доступ к данным и гибкость при добавлении новых единиц.

```

// Длина (базовая: метр)
{ "миллиметр", (QuantityType.Length, 0.001) },
{ "сантиметр", (QuantityType.Length, 0.01) },
{ "дециметр", (QuantityType.Length, 0.1) },
{ "метр", (QuantityType.Length, 1) },
{ "километр", (QuantityType.Length, 1000) },
{ "дюйм", (QuantityType.Length, 0.0254) },
{ "фут", (QuantityType.Length, 0.3048) },
{ "ярд", (QuantityType.Length, 0.9144) },
{ "миля", (QuantityType.Length, 1609.34) },
{ "парсек", (QuantityType.Length, 30856775814913672.79) },
{ "астрономическая единица", (QuantityType.Length, 149597870700) },
{ "световой год", (QuantityType.Length, 9460730472580800) },

// Масса (базовая: килограмм)
{ "миллиграмм", (QuantityType.Mass, 0.000001) },
{ "грамм", (QuantityType.Mass, 0.001) },
{ "килограмм", (QuantityType.Mass, 1) },
{ "центнер", (QuantityType.Mass, 100) },
{ "тонна", (QuantityType.Mass, 1000) },
{ "фунт", (QuantityType.Mass, 0.453592) },
{ "унция", (QuantityType.Mass, 0.0283495) },
{ "карат", (QuantityType.Mass, 0.0002) },

// Температура (базовая: Цельсий, коэффициенты фикт
{ "Цельсий", (QuantityType.Temperature, 1) },
{ "Фаренгейт", (QuantityType.Temperature, 1) },
{ "Кельвин", (QuantityType.Temperature, 1) },

// Скорость (базовая: м/с)
{ "м/с", (QuantityType.Speed, 1) },
{ "км/ч", (QuantityType.Speed, 0.277778) },
{ "мили/ч", (QuantityType.Speed, 0.44704) },
{ "км/с", (QuantityType.Speed, 1000) },
{ "узлы", (QuantityType.Speed, 0.514445) },

// Время (базовая: секунда)
{ "миллисекунда", (QuantityType.Time, 0.001) },
{ "секунда", (QuantityType.Time, 1) },
{ "минута", (QuantityType.Time, 60) },
{ "час", (QuantityType.Time, 3600) },
{ "день", (QuantityType.Time, 86400) },
{ "неделя", (QuantityType.Time, 604800) },
{ "год", (QuantityType.Time, 31536000) },

// Объем (базовая: литр)
{ "миллилитр", (QuantityType.Volume, 0.001) },
{ "литр", (QuantityType.Volume, 1) },
{ "галлон", (QuantityType.Volume, 3.78541) },
{ "куб.метр", (QuantityType.Volume, 1000) },

// Данные (базовая: байт)
{ "бит", (QuantityType.Data, 0.125) },
{ "байт", (QuantityType.Data, 1) },
{ "килобайт", (QuantityType.Data, 1024) },
{ "мегабайт", (QuantityType.Data, 1024 * 1024) },
{ "гигабайт", (QuantityType.Data, 1024 * 1024 * 1024) },

// Площадь (базовая кв.метр)
{ "кв.метр", (QuantityType.Square, 1) },
{ "кв.дециметр", (QuantityType.Square, 0.01) },
{ "кв.сантиметр", (QuantityType.Square, 0.0001) },
{ "гектар", (QuantityType.Square, 10000) },
{ "ар", (QuantityType.Square, 100) },
{ "акр", (QuantityType.Square, 4046.856) },

```

Рис. 1. Возможности конвертера величин.

Калькулятор поддерживает базовые арифметические операции (+, −, *, /), вычисление процентов (%), возведение в квадрат (x^2) и извлечение квадратного корня ($\sqrt{x}$). Обе модели работают независимо, но могут быть интегрированы в единый интерфейс для удобства пользователя. Такая архитектура обеспечивает модульность, простоту поддержки и масштабируемость приложения.

Калькулятор (рис. 2) обрабатывает операции в зависимости от их типа: бинарные (+, −, *, /, %) требуют двух чисел (a и b), выполняя между ними соответствующее арифметическое действие, а унарные (x^2 , $\sqrt{x}$) применяются только к первому числу (a), возводя его в квадрат или извлекая квадратный корень. При этом код включает проверки на исключительные ситуации: если пользователь пытается разделить на ноль, возвращается ошибка «Деление на ноль невозможно», а при извлечении корня из отрицательного числа — «Корень из отрицательного числа не определён». Это обеспечивает корректную работу программы и защиту от недопустимых вычислений.

Приложение включает проверку корректности ввода данных: если пользователь вводит текст вместо числа, система выводит сообщение «Ошибка: требуется числовое значение». Аналогично обрабатываются другие ошибочные сценарии: при делении на ноль выводится предупреждение «Деление на ноль невозможно», при попытке извлечь квадратный корень из отрицательного числа — «Ошибка: корень из отрицательного числа не определён», а при выборе несуществующей единицы измерения — «Указанная единица не поддерживается». Эти проверки обеспечивают стабильную работу программы и помогают пользователю быстро исправить ошибки ввода.

```
switch (op)
{
    case "+": result = a + b; break;
    case "-": result = a - b; break;
    case "*": result = a * b; break;
    case "/":
        if (b == 0) throw new DivideByZeroException();
        result = a / b;
        break;
    case "%": result = a * b / 100; break;
    case "x^2": result = a * a; break;
    case "sqrt":
        if (a < 0) throw new Exception("Нельзя извлечь корень из отрицательного");
        result = Math.Sqrt(a);
        break;
}
```

Рис. 2. Пример использования оператора case (switch) в коде.

```
if (!double.TryParse(convertInput.Text, out double input))
{
    MessageBox.Show("Ошибка: Введите число!");
    return;
}
```

Рис. 3. Скриншот экрана проверки на некорректный ввод.

Заключение. Разработанный универсальный конвертер величин и калькулятор на платформе WindowsForms представляет собой многофункциональный инструмент, объединяющий возможности преобразования физических величин и выполнения математических вычислений. Приложение поддерживает конвертацию единиц и измерения для восьми категорий, включая длину, массу, температуру и площадь, а также реализует базовые расширенные операции калькулятора, такие как вычисление процентов, возведение в квадрат и извлечение квадратного корня.

Основным преимуществом программы является интуитивно понятный интерфейс, который динамически адаптируется под выбранный режим работы: элементы управления конвертером скрываются при переходе к калькулятору, а для операций автоматически упрощается форма ввода. Алгоритмы конвертации обеспечивают высокую точность расчетов, включая нелинейные преобразования для температурных шкал. Система обработки ошибок предотвращает некорректные действия пользователя, такие как ввод нечисловых данных, деление на ноль или попытку конвертации несовместимых единиц.

Программа может быть полезна в образовательных целях, инженерных расчетах и повседневном использовании. Перспективы проекта включают добавление историй операций, экспорта результатов в форматы CSV/Excel, а также расширение списка поддерживаемых единиц измерения и математических функций. Дальнейшая оптимизация устройства повысит удобство работы с приложением для широкого круга пользователей.

Литература

1. Павловская Т.А. С#. Программирование на языке высокого уровня. – СПб.: Питер, 2009. – 432 с.
2. Техническая документация Microsoft. URL: <https://learn.microsoft.com/ru-ru/docs/> (дата обращения: 08.04.2025).
3. Ресурс METANIT.COM. URL: <https://metanit.com/sharp/tutorial/> (дата обращения: 15.04.2025).