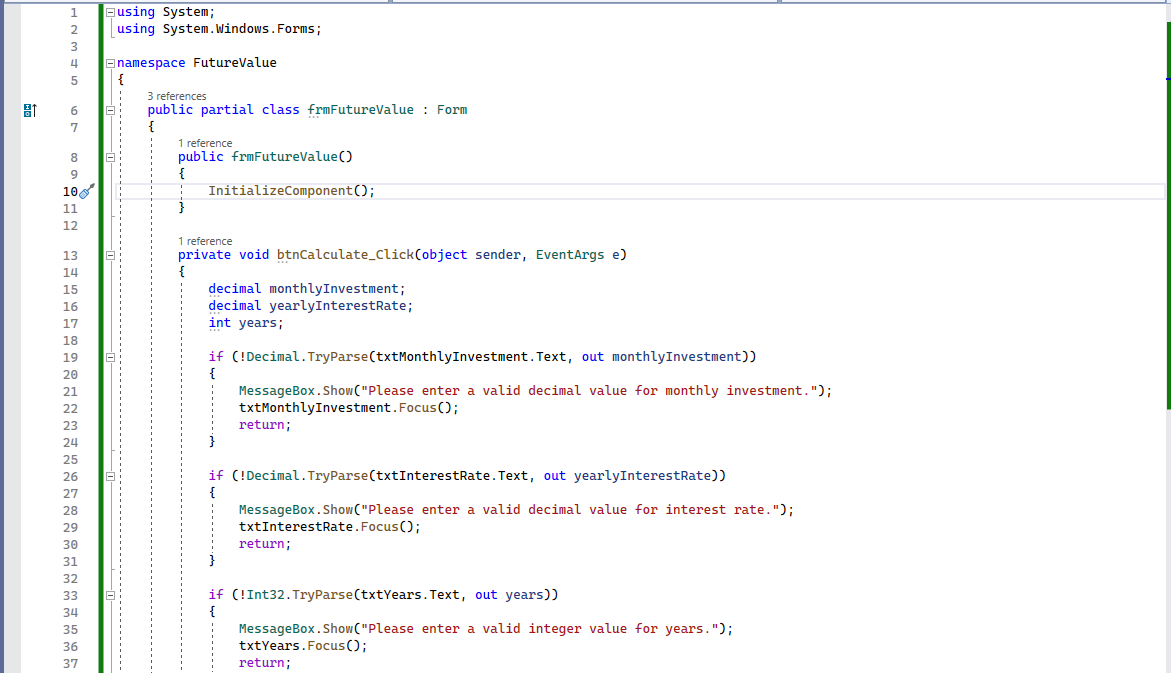
**The FutureMoneyValue application written in C#**

This C# code defines a Windows Forms application that calculates the future value of an investment based on user inputs for monthly investment, interest rate, and the number of years the investment will last.



Graphical user interface, text, application

Description automatically generated

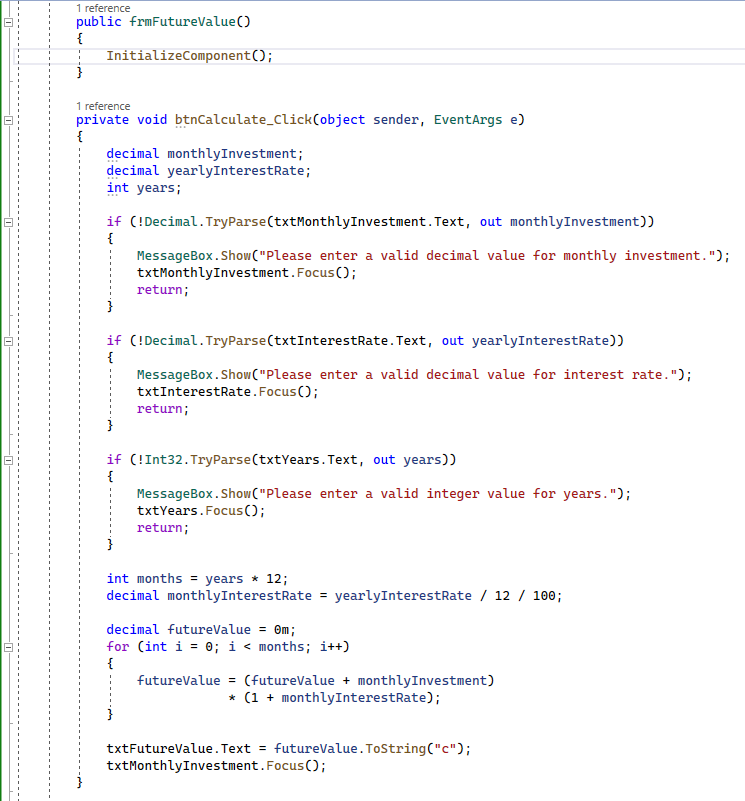
**The code imports the necessary namespaces:**

**Graphical user interface, text, application

Description automatically generated**

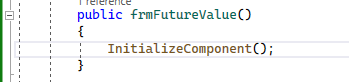
* System:
  + Provides core functionality such as input/output, string manipulation, and basic data types.
* System.Windows.Forms:
  + Contains classes for creating Windows-based applications that take full advantage of the rich user interface features available in the Microsoft Windows operating system.

The namespace FutureValue is a container for the application's class frmFutureValue, which inherits from the Form class. This is a common pattern for Windows Forms applications, as it provides a base class for the main form of the application.



**Within the frmFutureValue class, there are two primary methods:**

* public frmFutureValue():



* + This is the constructor for the frmFutureValue class.
  + It calls the InitializeComponent() method, which sets up the controls and properties of the form.
  + This method is typically auto-generated by the Windows Forms designer.
* private void btnCalculate\_Click(object sender, EventArgs e):

A picture containing shape

Description automatically generated

* + This is the event handler for the Click event of the btnCalculate button.
  + It contains the main logic for calculating the future value of the investment.

**Within the btnCalculate\_Click method, the code first declares three variables to hold the user input values:**

* monthlyInvestment:



* + The amount of money invested each month.
* yearlyInterestRate:



* + The annual interest rate of the investment.
* years:



* + The number of years the investment will last.

Next, it validates the user inputs by trying to parse each input as the appropriate data type using Decimal.TryParse() and Int32.TryParse(). If parsing fails, a message box is displayed to inform the user of the error, and focus is returned to the corresponding input textbox.

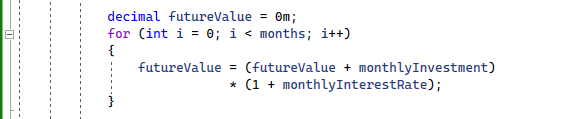
Graphical user interface, text

Description automatically generated

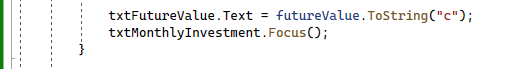
After validating the inputs, the code calculates the total number of months (months) and the monthly interest rate (monthlyInterestRate) based on the input values.



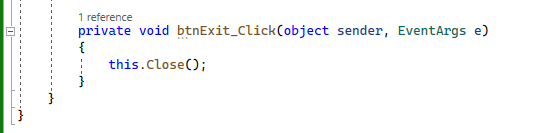
The future value of the investment is calculated using a loop that iterates for the total number of months. In each iteration, the futureValue variable is updated with the current investment and interest earned.



Finally, the calculated futureValue is formatted as a currency string and displayed in the txtFutureValue textbox. Focus is returned to the txtMonthlyInvestment textbox.



**The second method within the frmFutureValue class is:**

****

* private void btnExit\_Click(object sender, EventArgs e):



* + This is the event handler for the Click event of the btnExit button.



* + It simply closes the application form.

In summary, this code defines a Windows Forms application to calculate the future value of an investment based on user inputs. It validates the inputs, performs the calculation, and displays the result. The application also provides a button to exit the program.

**Program.cs**

This code snippet is the main entry point of a Windows Forms application named "FutureValue". It defines a static class called "Program" that sets up and runs the application. Here's a step-by-step explanation of the code:

Text

Description automatically generated

**Import the necessary namespaces:**

* System:



* + Provides core functionality such as input/output, string manipulation, and basic data types.
* System.Collections.Generic:



* + Contains interfaces and classes that define generic collections, which allow users to create strongly typed collections that provide better type safety and performance than non-generic collections.
* System.Linq:



* + Provides classes and interfaces that support queries using Language-Integrated Query (LINQ).
* System.Threading.Tasks:



* + Provides types that simplify the work of writing concurrent and asynchronous code.
* System.Windows.Forms:



* + Contains classes for creating Windows-based applications that take full advantage of the rich user interface features available in the Microsoft Windows operating system.

Define the namespace FutureValue as a container for the application's Program class.



Define a static class named Program. Since it is static, it cannot be instantiated, and all its methods and properties must also be static. This class acts as the container for the application's main entry point.



Add a summary comment for the Main method, which is the main entry point for the application. This comment provides documentation for the purpose of the method.

Add the [STAThread] attribute to the Main method. This indicates that the application's threading model is **Single Threaded Apartment (STA).** Windows Forms applications typically use the **STA threading model** because certain UI components require it for thread safety.

Text

Description automatically generated

**Define the Main method as the entry point for the application. Within this method, several important application settings are configured before the application is run:**

****

* Application.SetHighDpiMode(HighDpiMode.SystemAware):



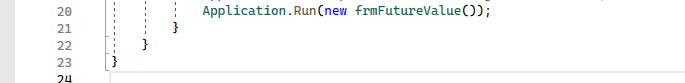
* + Sets the application's High DPI mode to "SystemAware".
  + This makes the application DPI-aware, allowing it to render properly on high-DPI displays.
* Application.EnableVisualStyles():



* + Enables visual styles for the application.
  + This makes controls on the form appear with the current Windows theme, resulting in a more modern look and feel.
* Application.SetCompatibleTextRenderingDefault(false):



* + Sets the default text rendering mode for the application.
  + By setting it to false, the application uses the newer GDI+ text rendering, which typically provides better quality and performance.
* Application.Run(new frmFutureValue()):



* + Starts the application event loop and displays the main form.
  + The new frmFutureValue() expression creates an instance of the frmFutureValue form, which was defined in the previous code snippet you provided.

In summary, this code snippet defines the main entry point for a Windows Forms application called "FutureValue". It sets up the application with appropriate settings for High DPI mode, visual styles, and text rendering. Then, it starts the application event loop and displays the main form, frmFutureValue.

This code snippet is part of the Windows Forms application named "FutureValue". It defines a partial class frmFutureValue, which contains the designer-generated code for the form's controls and layout. This class is responsible for setting up the user interface and initializing the form's components.

**frmFormFutureValue.Designer.cs**

**Here's a step-by-step explanation of the code:**

**Text

Description automatically generated**

Define the namespace FutureValue as a container for the application's frmFutureValue partial class.



Define a partial class named frmFutureValue. This partial class is combined with the previously defined frmFutureValue class to create the full class definition. This separation allows the designer-generated code to be kept separate from the custom code.

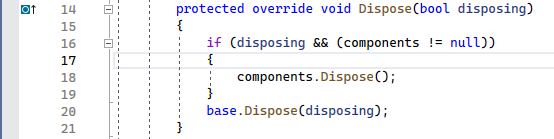
Text

Description automatically generated with low confidence

Declare a private field named components of type System.ComponentModel.IContainer. This field holds references to the form's components, which are required for the designer support and for disposing of resources when the form is closed.



Define the Dispose method, which is responsible for releasing unmanaged resources and disposing of managed resources when the form is closed. This method takes a single parameter disposing, which is a boolean indicating whether to dispose of managed resources. The method checks if disposing is true and if components is not null, then it calls the Dispose method on the components. Finally, the method calls the Dispose method on the base class (Form).

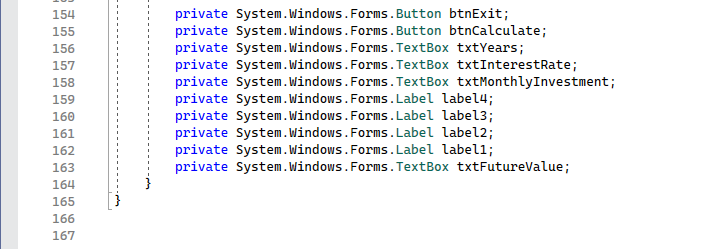


Add a region directive named Windows Form Designer generated code. This directive is used to group and organize the designer-generated code.



Define the InitializeComponent method, which is responsible for initializing the form's controls and properties. This method is called in the constructor of the frmFutureValue class.

Create instances of the controls (buttons, textboxes, and labels) and set their properties, such as location, size, name, and text.



* Register event handlers for the click events of the btnExit and btnCalculate buttons.
* Define properties of the form, such as size, name, start position, and text.
* Add the controls to the form's Controls collection.
* Call the SuspendLayout method before setting the control properties to optimize performance by temporarily suspending the layout logic.
* Call the ResumeLayout and PerformLayout methods after setting the control properties to resume the layout logic and apply the layout changes.
* Close the region directive for the Windows Form Designer generated code.
* Declare the private fields for each control on the form. These fields hold references to the controls and can be used in the custom code to interact with the controls.

In summary, this code snippet defines a partial class frmFutureValue that contains the designer-generated code for setting up the form's user interface. It initializes the form's controls and properties, registers event handlers, and adds the controls to the form. The partial class is combined with the previously defined frmFutureValue class to create the full class definition for the form.