Programowanie obiektowe.

Obiekt

Obiekt to dowolny element, który możemy wydzielić i którym możemy manipulować. W terminologii informatycznej obiekt to samodzielna jednostka zawierająca zarówno dane, jak i kod zorganizowany w postaci podprogramów i funkcji, umożliwiający dostęp do tych danych lub ich modyfikację.

Klasa

Klasa definiuje funkcję obiektu, jego cechy charakterystyczne i tożsamość. Klasę można przyrównać do szablonu lub wzorca, według którego budowany jest obiekt.

Składnia klasy:

```
[Shadows]
[Mustinherit | Notlnheritable]
[Public | Private | Protected | Friend | Protected Friend]
Class NazwaKlasy
[Inherits NazwaKlasyBazowej]
[Implements InterfejsA, InterfejsB, … InterfejsN}
[kod klasy]
End Class
```

Interfejsy

Interfejs jest niczym klasa, która posiada modyfikator Mustinherit i której każda ze składowych ma modyfikator MustOverride. Wewnątrz interfejsu można zadeklarować metody, właściwości oraz zdarzenia. Nazwa każdego interfejsu rozpoczyna się od litery **I**.

Składnia interfejsu:

```
[Shadows]
 [Public | Private | Protected | Friend | Protected Friend]
Interface NazwaInterfejsu
 [Inherits InterfejsA, InterfejsB, … InterfejsN}
[kod interfejsu]
End Interface
```

Metody

Metoda to zestaw instrukcji wykonywanych jako całość. Metoda może zwracać jedną wartość lub tablicę (zbiór) wartości (Funkction). Można również utworzyć metodę, która nie zwraca żadnej wartości (Sub).

Składnia instrukcji Sub:

```
[Overloads | Shadows | Shared]
[Overrides | Overridable | NotOverridable | MustOverride]
[Public | Private | Protected | Friend | Protected Friend]
Sub NazwaMetody() [Implements Interfejs.Składowa]
[kod metody]
[Exit Sub]
[kod metody]
End Sub
```

Składnia instrukcji function:

```
[Overloads | Shadows | Shared]
[Overrides | Overridable | NotOverridable | MustOverride]
[Public | Private | Protected | Friend | Protected Friend]
Sub NazwaMetody() [Implements Interfejs.Składowa]
[kod metody]
[Exit Function]
[kod metody]
End Sub
```

Zmienne

Zmienna nazywana również polem służy do przechowywania wartości. W czasie działania programu wartość przechowywana w zmiennej może się zmieniać, chyba, że zmienna posiada modyfikator ReadOnly.

Składnia instrukcji Dim:

[Shared]
[Shadows]
[ReadOnly]
[Public I Pnvate | Protected | Friend | Protected Friend]
Dim NazwaZmiennej As TypDanych [= WartośćPoczątkowa]

Jeżeli przed instrukcją Dim umieścisz modyfikator, edytor kodu usuwa instrukcję Dim. Aby zadeklarować zmienną wewnątrz metody, nie należy używać modyfikatorów, a tylko instrukcji Dim. Zmienna zadeklarowana wewnątrz metody jest dostępna tylko wewnątrz niej.

Składowa instancyjna

Aby uzyskać dostęp do składowej instancyjnej, najpierw należy utworzyć instancję (egzemplarz) klasy, która ją udostępnia.

Przykład udostępniania:

Dim [WithEvents] InstancjaKlasy As NazwaKlasy = New NazwaKlasy()
InstancjaKlasy.SkładowaInstancyjna

Składowa współdzielona

Składowa współdzielona nie jest związana z konkretną instancją (egzemplarzem) klasy.

Przykład udostępniania:

NazwaKlasy.SkładowaWspółdzielona

Przykład definiowania:

Public **Shared** Sub SkladowaWspóldzielona()

Własności

Właściwość, podobnie jak metoda, to zestaw instrukcji wykonywanych jako całość. Metodę można traktować jako akcję, natomiast właściwość jako atrybut (cechę) klasy.

Składnia instrukcji Property:

[Overloads | Shadows | Shared]
[Overrides | Overridable | NotOverridable | MustOverride]
[WriteOnly | ReadOnly]
[Public | Private | Protected | Friend | Protected Friend]
Property NazwaWlaściwości() As Typowanych [Implements Interfejs.Składowa]
Set(ByVal wartość As TypDanych)
  [kod właściwości]
End Set
Get
  [kod właściwości]
End Get
End Property

Zdarzenia i delegaty

Zdarzenie to akcja, na skutek której obiekt wysyła określoną informację. Zdarzenia mogą być generowane przez działania użytkownika lub uruchamiane przez logikę programu.

Delegaty to obiekty typu referencyjnego, który stanowi interfejs metody o odpowiedniej sygnaturze oraz zwracanym typie.

Stałe

Stała służy do reprezentowania wartości. Stała, od początku do końca działania programu, reprezentuje zawsze tę samą wartość

Składnia:

```
[Shadows]
[Public | Priivate | Protected | Friend | Protected Friend]
Const NazwaStałej As TypDanych = Wartość
```

Enumaracje

Enumeracja to zbiór powiąnych ze sobą stałych.

Składnia:

```
[Shadows]
[Public | Priivate | Protected | Friend | Protected Friend]
Enum NazwaEnumeracji As TypDanych
  Elementl [= Wartości1]
  Element2 [= Wartości2]
 . . .
   ElementN [= WartościN]
End Enum
```

Tablice

Tablica służy do przechowywania zbioru wartości. To, jakie wartości można przechowywać w tablicy, zależy od jej typu danych.

Składnia:

Tablica jednowymiarowa

```
Dim NazwaTablicy As TypDanych()
NazwaTablicy = New TypDanych(IndeksGórny) {}
```

Tablica wielowymiarową

```
Dim NazwaTablicy As TypDanych (,)
NazwaTablicy = New TypDanych(IGWP, IGWD) {}
```

Modyfikatory instrukcji

Modyfikator	Metoda	Własności	Zmienna	Klasy	Opis
Overload	V	V			Właściwość/Metoda posiada więcej niż jedną wersję. Modyfikator Overloads jest opcjonalny — można utworzyć właściwość/metodę o wielu wersjach bez korzystania z tego modyfikatora.
Shadows	V	V	V	V	Właściwość/Metoda/Zmienna/Klasa klasy potomnej zasłania składową, która ma identyczną nazwę, zdefiniowaną w klasie bazowej. Aby zasłonić właściwość/metodę/zmienną nową właściwością/metodą/zmienną, zestaw parametrów i typ danych nowej właściwości/metody może się różnić od zestawu parametrów i typu danych właściwości zasłanianej.
Shared	V	V	V		Właściwość/Metoda/Zmienna współdzielona przez wszystkie instancje klasy.
Overrides	V	V			Właściwość/Metoda klasy potomnej zmienia implementację właściwości/metody klasy bazowej, Jedna i druga właściwość/metoda ma taką samą nazwę. Aby zmienić właściwość/metodę, liczba parametrów, typy danych parametrów oraz typ danych nowej właściwości/metody musi zgadzać się z liczbą parametrów, typami danych parametrów oraz typami danych właściwości/metody zmienianej.
Overridable	V	V			Właściwość/Metoda może zostać zmieniona przez właściwość/metodę, która ma identyczną nazwę, zdefiniowaną w klasie potomnej.
NotOverridable	V	V			Właściwość/Metoda nie może zostać zmieniona przez właściwość/metodę, która ma identyczną nazwę, zdefiniowaną w klasie potomnej
MustOverr1de	V	V			Właściwość/Metoda oprócz nagłówka nie posiada kodu (nie jest zaimplementowana). Właściwość/Metoda ta musi zostać zaimplementowana w klasie potomnej, aby ta klasa mogła działać prawidłowo.
WriteOnly		V	V		Wartość właściwości można zmieniać, ale nie można jej odczytywać.
ReadOnly			V		Wartość właściwości/zmiennej można odczytywać, ale nie można jej zmieniać
Mustinherit				V	Aby uzyskać dostęp do składowych instancyjnych klasy, trzeba najpierw utworzyć klasę, która ją dziedziczy. Nie można utworzyć instancji klasy mającej modyfikator Mustinherit.
Notinheritable				V	Klasa nie może być dziedziczona, co oznacza, że nie można jej używacywać jako klasy bazowej dla innych klas.

Modyfikatory dostępu

Modyfikator	Metoda	Zmienna	Własność	Opis
Public	V	V	V	Właściwość/Metoda/Zmienna jest dostępna bez żadnych ograniczeń.
Private	V	V	V	Właściwość/Metoda/Zmienna jest dostępna tylko wewnątrz klasy, w której została utworzona.
Protected	V	V	V	Właściwość/Metoda/Zmienna jest dostępna wewnątrz klasy, w której została utworzona, oraz wewnątrz klasy potomnej.
Friend	V	V	V	Właściwość/Metoda/Zmienna jest dostępna tylko wewnątrz programu, w którym została utworzona.
Protected Friend	V	V	V	Właściwość/Metoda/Zmienna jest dostępna z ograniczeniami wynikającymi z połączenia modyfikatorów Protected i Friend.

Instrukcje konfiguracyjne

Option Compare Binary -sprawia, że tekst jest porównywany metodą binarną co oznacza m.in., że Visual Basic .NET odróżnia małe litery od wielkich.
Option Explicite On sprawia, że Visual Basic .NET sprawdza, czy każda zmienna przed użyciem została zadeklarowana. Dzięki deklarowaniu oprawia się czytelność kodu.
Option Stricte On sprawia, że Visual Basic .NET sprawdza, czy wszystkie zmienne i parametry zostały zadeklarowane z podaniem ich typów.

Podstawy

Metoda **Main**, punkt startowy programu, korzysta z bibliotek *System* oraz *System. Windows.Forms*.
Przykład:

Public Shared Sub Main() [kod metody Main] End Sub	Public Shared Sub Main(ByVal P As String()) [kod metody Main] End Sub
Public Shared Function MainO As Integer [kod metody Main] End Sub	Public Shared Function Main(ByVal P As StringO) As Integer [kod metody Main] End Sub

Metoda **MyBase**, umożliwia dostęp do klasy bazowej z klasy potomnej