

Re: Collection/list problem

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2008-07/msg00822.html>

- *From:* Lev Blaivas <Lev.Blaivas@xxxxxxxx>
 - *Date:* Thu, 24 Jul 2008 07:05:08 -0700 (PDT)
-

On Jul 24, 11:54 am, "Clive Lumb"
<clumb2@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

Yay! it works!

I would recommend reading once the language specification to clear out the basics...

Unfortunately I have the French version which is dire, for example it says that objects in a collection **should** not be of the same type (instead of **need** not).

"Patrice" <<http://www.chez.com/scribe/>> a écrit dans le message de news: 7A8AF0C2-88AC-4B3D-9FBC-24835CAF4...@xxxxxxxxxxxxxxxxxxxx

You need to add MyClass=New MyThing inside the loop.

Here you always adding the same object to the collection. Even once the object is added to the collection you can change the object properties (basically an object is a pointer actual data).

I would recommend reading once the language specification to clear out the basics...

--
Patrice

Re: Collection/list problem

"Clive Lumb" <clumb2@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> a écrit dans
le
message de groupe de discussion :
488839a0\$0\$21312\$426a7...@xxxxxxxxxxxxxxxxxxxx

Hi,

I'm having a problem while adding objects to a collection.
(VB.net 2005)
I am reading data in from a file to a temporary variable (eg
MyClass) and
then adding it to a collection.
When I then enumerate the collection, all the entries are
equal to the
last
one added.
The same thing happens with lists and with a generic list(of
type).
Obviously the collection or list is storing a reference to the
variable,
rather than the values – hence all the items refer to the same
variable.

I am probably missing something really basic – please be
patient :-)

Many TIA

example:

```
Public Class MyThing
    name as String
    Value as integer
end class
```

```
Dim MyClass as New MyThing
Dim MyCollection as New Collection
```

Re: Collection/list problem

```
'open file
'do
  ' read in name, value
  MyClass.name=NameFromFile
  MyClass.value=ValueFromFile
  MyCollection.Add(MyClass)
'loop to end of file
'close file
```

Here is a great article about reference and value types;

<http://www.ondotnet.com/lpt/a/2789>

explains in detail the difference between value and reference types

.