Structure (Struct) in C

In C #, a structure is a data type. It helps you create a single variable that keeps relevant data of diverse data types. The keyword struct in C # is used to create a structure.

In C #, a structure is a data type. It helps you create a single variable that keeps relevant data of diverse data types. The keyword struct in C # is used to create a **structure** .

Structures are used to represent a record. Suppose you want to keep track of books in a library. You may want to track the following attributes of each book:

1. Book title
2. Author
3. Kind
4. ID (book code)

Defining structure in C #

To define the structure, you must use the struct command. The struct statement defines a new data type, with more than one member in your program.

The following example is how you declare the Book structure:

```
struct Books { public string  ten_sach ; public string  tac_gia ; public string
```

The following program illustrates how to use the above structure in C #:

```
using System ; //cau truc book struct Book { public string  ten_sach ; public s
```

If you do not use the **Console.ReadKey ()** command; then the program will run and finish (so fast that you can not see the results). This command allows us to see the results more clearly.

Characteristics of structure in C #

Above, you used a simple Books structure. Structures in C # are quite different from the traditional structure in C or C ++. The structure in C # has the following characteristics:

The structure can have methods, fields, indexers, properties, operator methods, and events.

The structure may have defined constructors, but not destructors. However, you cannot define a default constructor for a structure. The default constructor is automatically defined and cannot be changed.

Unlike Classes, the structure cannot inherit from another structure or class.

The structure cannot be used as a basis for structures or other classes.

A structure can deploy one or more interfaces.

Structure members cannot be defined as abstract, virtual, or protected.

When you create a Struct object by using the **new** operator, it takes the created object and the appropriate constructor is called. Unlike the class, the structure can be initialized without using the new operator.

If the new operator is not used, then the fields are not assigned and the object cannot be used until all fields are initialized.

Differentiate Class and Structure in C #

Classes and Structures in C # have some basic differences:

Classes are reference types, and structures are value types.

The structure does not support inheritance.

The structure has no default constructor.

From the above points, we rewrite the example above:

```
using System ; struct Book { private string  ten_sach ; private string  tac_gia
```

If you do not use the Console.ReadKey () command; then the program will run and finish (so fast that you can not see the results). This command allows us to see the results more clearly.

Follow tutorialspoint

Previous lesson: String (String) in C #

Next lesson: Class (Class) in C #

You finished reading the article "**Structure (Struct) in C #**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.