Records

Cloning and expression

Cloning

Previously, we learned that a type provides immutability by default. Without adding an explicit setter to the property, we cannot modify any value of the instance.

However, the combined with new keyword can perform a **nondestructive mutation**. The new expression makes a clone of the record instance with specified properties and fields modified.

The part after the keyword is just the object initializer which is called on the nearly created clone object. We can also create a copy of the object without changing the properties.

Shallow copy

The result of a expression is a **shallow copy**. It means that for each reference property, only the reference to an instance is copied. Both the original record and the cloned end up with references to the same instance on the heap.

Cloning

The cloning functionality of the record is provided by the automatically generated
method and a constructor, which rewrites values of the properties to the new instance.

Custom cloning

If you need different copy behavior, you can write your own copy constructor. If you do that, the compiler won't generate a second one.

Reserved name

You can't override the method or name any member with this name. The real name of the clone method is compiler-generated, but still, this action is forbidden.