Records

`ToString()` **formatting**

ToString **method**

A **record** type provides a helpful **ToString()** formatting that makes the object human-readable. By default, it applies **$"{propertyName} = {propertyValue.ToString()},"** pattern on all properties

```csharp
public record SpeakerDTO(string FirstName, string LastName, params string[] Sessions);

public static void Main()
{
    var speaker = new SpeakerDTO("Szymon", "Kulec", "Pro C# 9 Expert for everyone", "How to become Async Expert");
    Console.WriteLine(speaker);
    // prints
    // SpeakerDTO { FirstName = Szymon, LastName = Kulec, Sessions = System.String[] }
}
```

ToString **internals**

```csharp
public class SpeakerDTO : IEquatable<SpeakerDTO>
{
  public override string ToString()
  {
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.Append("SpeakerDTO");
    stringBuilder.Append(" { ");
    if (PrintMembers(stringBuilder))
    {
      stringBuilder.Append(" ");
    }
    stringBuilder.Append("}");
    return stringBuilder.ToString();
  }

  [...]
}
```

ToString **internals**

```csharp
public class SpeakerDTO : IEquatable<SpeakerDTO>
{
  [...]

  protected virtual bool PrintMembers(StringBuilder builder)
  {
    builder.Append("FirstName");
    builder.Append(" = ");
    builder.Append(FirstName);
    builder.Append(", ");
    builder.Append("LastName");
    builder.Append(" = ");
    builder.Append(LastName);
    builder.Append(", ");
    builder.Append("Sessions");
    builder.Append(" = ");
    builder.Append(Sessions);
    return true;
  }
}
```

Custom formatting

We can adjust and tune the default formatting by providing or overriding the method (depending on the inheritance hierarchy). For example, to make `SpeakerDTO` printable with session titles, let's implement the `private bool PrintMembers(StringBuilder builder)` method.

Our record is not `sealed`, so the compiler requires from us to make the method `protected` and `virtual`, to allow any child record to also customize its formatting.

Custom formatting

```csharp
public record SpeakerDTO(string FirstName, string LastName, params string[] Sessions)
{
  protected virtual bool PrintMembers(StringBuilder builder)
  {
    builder.Append("FirstName = ");
    builder.Append(FirstName);
    builder.Append(", LastName = ");
    builder.Append(LastName);
    foreach (var session in Sessions)
    {
        builder.Append(", Session title: = ");
        builder.Append(session);
    }
    return true;
  }
}

// SpeakerDTO { FirstName = Szymon, LastName = Kulec,
//    Session title: = Pro C# 9 Expert for everyone, Session title: = How to become Async Expert }
```