06

Corrigindo uma refatoração problemática

Você está desenvolvendo uma aplicação de cartão de ponto para a área de recursos humanos.

Num certo momento, toda a aplicação está funcionando, então você resolve analisar o programa e possivelmente melhorar a qualidade do código.

A classe `CartaoDePonto` está como na listagem abaixo:

```
class CartaoDePonto
{
    // Aqui vão alguns métodos de apontamento de horas muito úteis...

    public CartaoDePonto()
    {
        var data = DateTime.Today;
        var ultimoDiaDoMes = UltimoDiaDoMes(data);
        var primeiroDiaDoMes = PrimeiroDiaDoMes(data);
        var ehFimDeSemana = EhFimDeSemana(data);
    }

    private DateTime PrimeiroDiaDoMes(DateTime data)
    {
        return new DateTime(data.Year, data.Month, 1);
    }

    private DateTime UltimoDiaDoMes(DateTime data)
    {
        return new DateTime(data.Year, data.Month, DateTime.DaysInMonth(data.Year, dat
    }

    private bool EhFimDeSemana(DateTime data)
    {
        return data.DayOfWeek == DayOfWeek.Saturday
            || data.DayOfWeek == DayOfWeek.Sunday;
    }
}
```

Você então percebe que a classe `CartaoDePonto` está acumulando *métodos estrangeiros*: `UltimoDiaDoMes`, `PrimeiroDiaDoMes` e `EhFimDeSemana`. Você decide então criar uma extensão para conter esses métodos.

Após a refatoração, o código fica como a listagem abaixo:

```
namespace Programa
{
    class Exemplo
    {
        public Exemplo()
        {
            var data = DateTime.Today;
```

```
            var ultimoDiaDoMes = data.UltimoDiaDoMes();
            var primeiroDiaDoMes = data.PrimeiroDiaDoMes();
            var ehFimDeSemana = data.EhFimDeSemana();
        }
    }

    static class DateTimeExtensions
    {
        public static DateTime PrimeiroDiaDoMes(DateTime data)
        {
            return new DateTime(data.Year, data.Month, 1);
        }

        public static DateTime UltimoDiaDoMes(DateTime data)
        {
            return new DateTime(data.Year, data.Month, DateTime.DaysInMonth(data.Year,
        }

        public static bool EhFimDeSemana(DateTime data)
        {
            return data.DayOfWeek == DayOfWeek.Saturday
                || data.DayOfWeek == DayOfWeek.Sunday;
        }
    }
}
```

Mas infelizmente o Visual Studio exibe as seguintes mensagens de erro:

```
Error    CS1061    'DateTime' does not contain a definition for 'UltimoDiaDoMes'
and no extension method 'UltimoDiaDoMes' accepting a first argument of type 'DateTime'
could be found (are you missing a using directive or an assembly reference?)

Error    CS1061    'DateTime' does not contain a definition for 'PrimeiroDiaDoMes'
and no extension method 'PrimeiroDiaDoMes' accepting a first argument of type 'DateTim
could be found (are you missing a using directive or an assembly reference?)

Error    CS1061    'DateTime' does not contain a definition for 'EhFimDeSemana'
and no extension method 'EhFimDeSemana' accepting a first argument of type 'DateTime'
could be found (are you missing a using directive or an assembly reference?)
```

Qual a causa mais provável do problema?

Selecione uma alternativa

A — A classe está sem a visibilidade **pública**

B — A classe está indevidamente marcada como **static**

C — Os métodos da classe de extensão estão indevidamente marcados como **static**.

D Está faltando o modificador **this** nos parâmetros `DateTime data` dos métodos de extensão.