

Flyweight em C#

É o pattern que diminui o consumo de memória ao compartilhar dados entre objetos similares. Normalmente é usado com coleções pequenas de objetos.

Quando usar:

- Quando você tem muitos objetos em memória;
- Quando você pode compartilhar estado entre objetos;
- Quando você quer economizar memória.

Como usar:

- Criar as classes **Flyweight**, que implementam **IFlyweight**;
- Criar uma class **FlyweightFactory**, que possui uma coleção de **IFlyweight**;
- Implementar as operações (**Operation**);
- Criar um construtor que inicialize a coleção de **IFlyweight**.

O **Flyweight** seria parecido com o **Singleton** se você, de algum modo, reduzisse todos os estados de objetos compartilhados para apenas um objeto flyweight. Mas há duas mudanças fundamentais entre esses padrões:

- Deve haver apenas uma única instância singleton, enquanto que uma classe *flyweight* pode ter múltiplas instâncias com diferentes estados intrínsecos.
- O objeto *singleton* pode ser mutável. Objetos **flyweight** são imutáveis.

Prós

Você pode economizar muita RAM, desde que seu programa tenha muitos objetos similares.

Contras

Você pode estar trocando RAM por ciclos de CPU quando parte dos dados de contexto precisa ser recalculada cada vez que alguém chama um método flyweight.

O código fica muito mais complicado. Novos membros de equipe sempre se perguntarão por que o estado de uma entidade foi separado de tal forma.