Centro Universitário UniBH
Ciência da Computação
Práticas de Programação
Professor: Lucas Schmidt

Aula Prática - 12 Padrões Estruturais - Proxy

Intenção

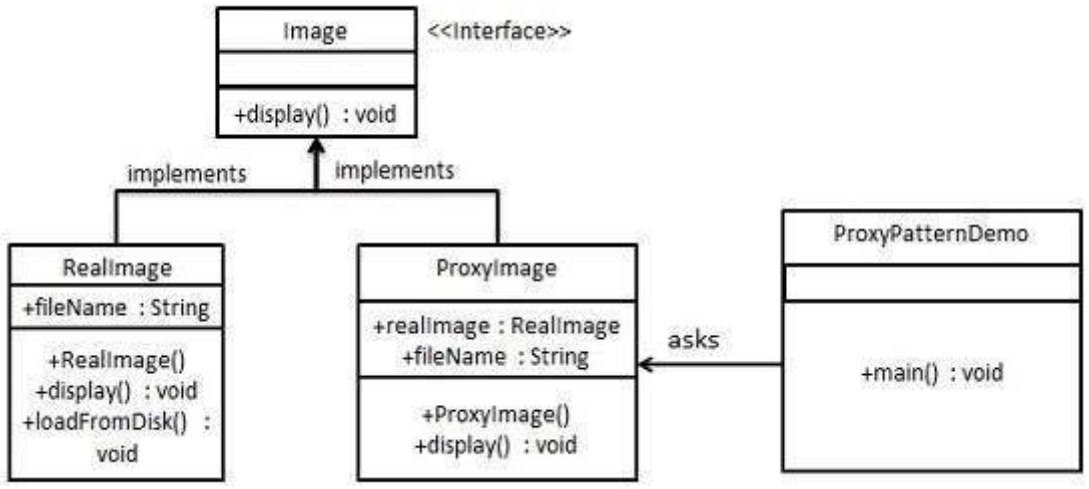
Prover um representante ou ponto de acesso que controle o acesso a um objeto.

Usar este padrão quando…

- Precisar de um acesso mais versátil a um objeto do que um ponteiro:
  - Remote proxy (acesso remoto);
  - Virtual proxy (exemplo da imagem);
  - Protection proxy (controla acesso).

Vantagens e desvantagens

- Adiciona um nível de separação:
  - Transparência na execução de ações de carregamento de objetos.



Passo 1
Crie uma interface.

Image.java

```
public interface Image {
   void display();
}
```

Passo 2
Crie classes concretas implementando a mesma interface.

RealImage.java

```
public class RealImage implements Image {

   private String fileName;

   public RealImage(String fileName){
      this.fileName = fileName;
      loadFromDisk(fileName);
   }

   @Override
   public void display() {
      System.out.println("Displaying " + fileName);
   }

   private void loadFromDisk(String fileName){
      System.out.println("Loading " + fileName);
   }
}
```

ProxyImage.java

```
public class ProxyImage implements Image{

   private RealImage realImage;
   private String fileName;

   public ProxyImage(String fileName){
      this.fileName = fileName;
   }

   @Override
   public void display() {
      if(realImage == null){
         realImage = new RealImage(fileName);
      }
      realImage.display();
   }
}
```

Passo 3
Use o ProxyImage para obter o objeto da classe RealImage quando necessário.

ProxyPatternDemo.java

```
public class ProxyPatternDemo {

   public static void main(String[] args) {
      Image image = new ProxyImage("test_10mb.jpg");

      //image will be loaded from disk
      image.display();
      System.out.println("");

      //image will not be loaded from disk
      image.display();
   }
}
```

Passo 4

Teste sua implementação!