

Ian Carvalho – 7959725

Marcelo Rabello Rossi – 5121901

Sexto Mini-Exercício Programa

Relatório

O presente trabalho teve como objetivo a aplicação dos *design patterns* *Factory* e *Strategy* em projeto de um Media Center simplificado.

Primeiramente o Media Center foi desenvolvido sem nenhuma preocupação com padrões. O código está disponível na pasta 'raw', entregue juntamente com o diagrama de classes UML que descreve a estrutura das classes dessa primeira etapa (Figura 1). O programa é funcional e pode ser executado a partir do módulo main.py. Considerando o objetivo do trabalho, poucas funções foram implementadas. Será possível adicionar algumas mídias ao Media Center e executá-las, em linha de comando.

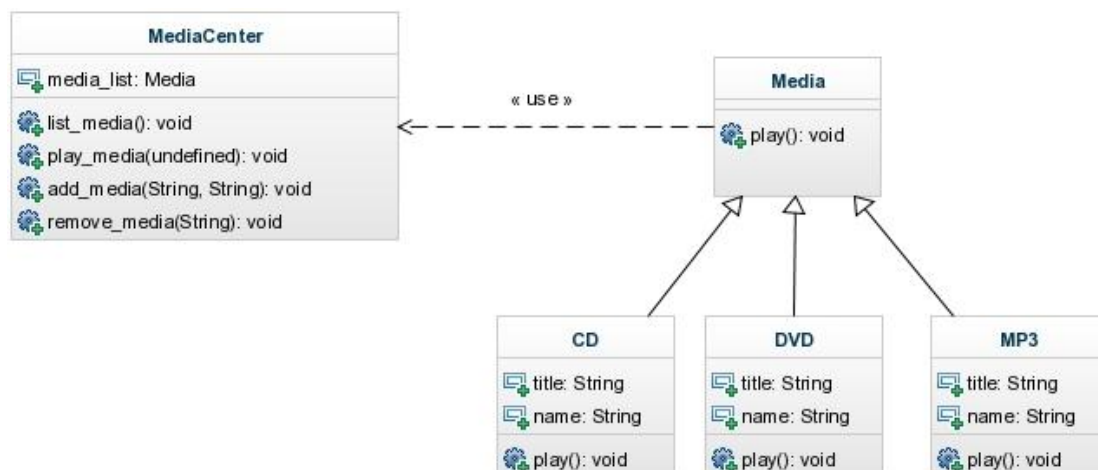


Figura 1. Diagrama de classes UML para o projeto Media Center antes da implementação de *design patterns*.

A partir disso, o projeto foi repensado de acordo com os *design patterns* *Factory* e *Strategy*. O código está disponível na pasta 'design_patterns', juntamente com seu diagrama de classes UML (Figura 2).

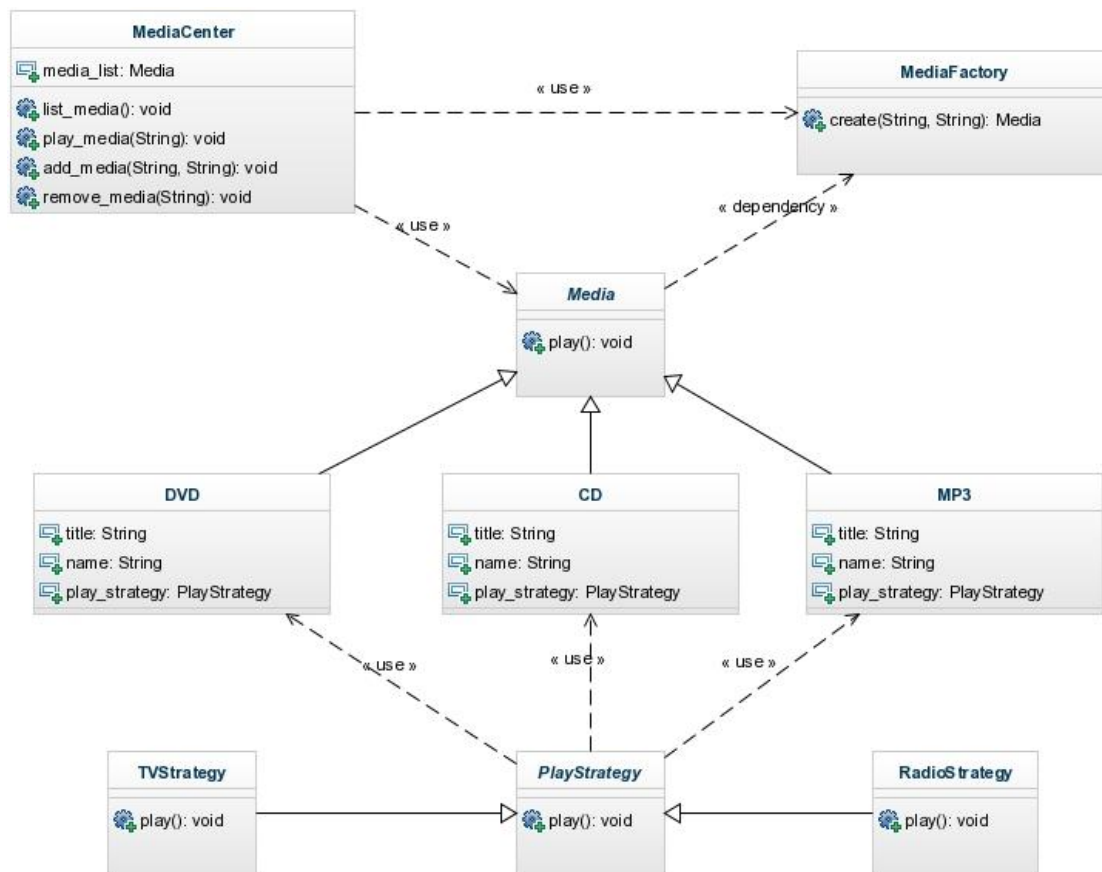


Figura 2. Diagrama de classes UML para o projeto Media Center após a implementação dos padrões *Factory* e *Strategy*.

Apesar do aparente ganho de complexidade quando comparados os diagramas de classe, a implementação do padrão *Factory* resultou em maior modularização e encapsulamento do código, de modo a facilitar inserções ou exclusões futuras de formatos de mídia.

A implementação da padrão *Strategy* eliminou a repetição de código necessária para executar diversos formatos de mídia. Nesse caso, mídias que utilizam a mesma estratégia de execução (ex. CD e MP3 podem ser executados no rádio). Os formatos de mídia adicionados futuramente poderão se utilizar dessas estratégias para serem executadas.

Esses ganhos podem parecer pequenos em projetos do tamanho desse Media Center, mas passam a ser cruciais em projetos maiores, nos quais essas 'más práticas' podem provocar grandes transtornos a cada nova implementação ou manipulação do código.