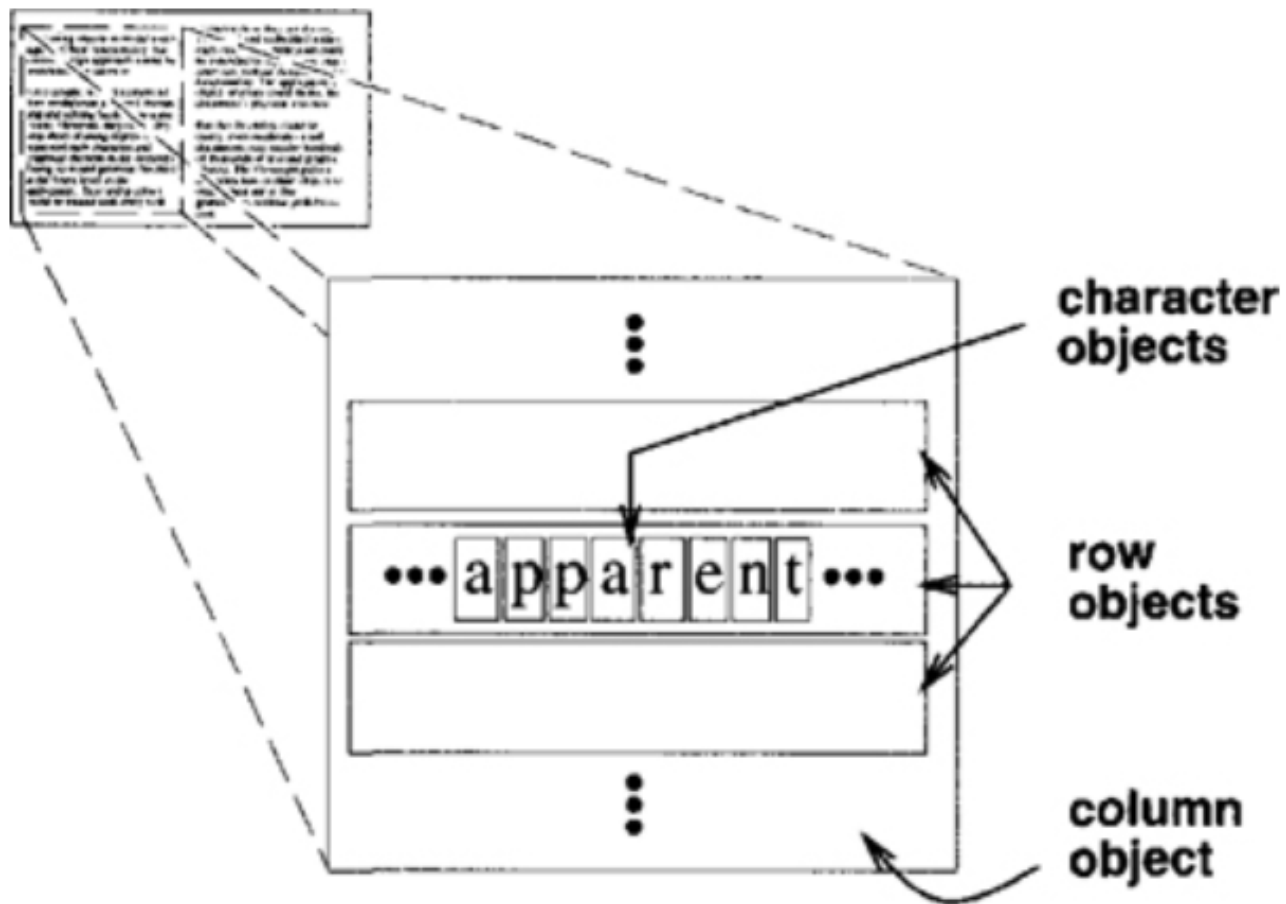


Flyweight

Análise e Desenvolvimento de Sistemas - 6º Período

Cenário

- Imagine que estamos construindo um editor de texto usando orientação a objetos
- Cada elemento da janela do editor pode ser considerado um objeto
- Para **modularizar** ainda mais nossa aplicação, podemos imaginar que cada **caracter** é um objeto



Cenário

- O objeto **character** pode ter algumas propriedades, como: linha, coluna, fonte
- Qual o problema nesse cenário? **A quantidade de objetos criados!**
- Imagine seu artigo de TCC, onde cada character é um objeto instanciado!

Cenário

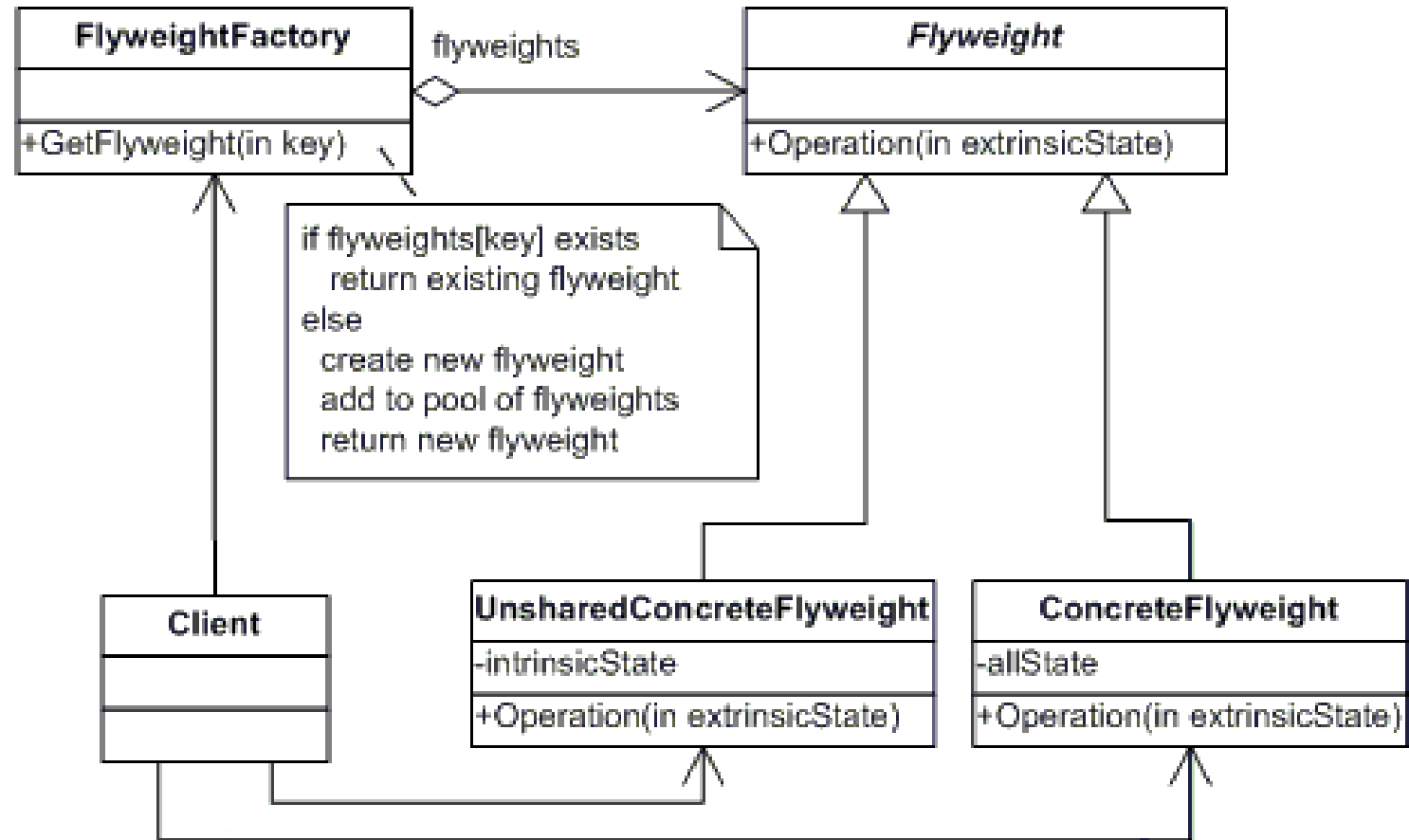
- Temos muitos objetos que compartilham das mesmas características sendo criados
- Isso pode ocasionar num **consumo grande de recursos como memória e um run-time overhead.**

Flyweight

- Otimizar recursos e memória, **compartilhando** objetos que possuem partes em comum, enquanto **mantém o estado exclusivo separado**.
- Usado quando grande número de objetos semelhantes precisa ser criado
- Cada objeto compartilha parte de seu estado com os objetos semelhantes
- Esse compartilhamento permite economizar memória e reduzir sobrecarga associada a criação de objetos

Conceitos Fundamentais

- **Estado intrínseco:** Estado interno ao contexto do flyweight. Compartilhado entre os objetos;
- **Estado extrínseco:** Estados que não são compartilhados entre os objetos
- **Flyweight factory:** Gerencia os objetos. Garante que os objetos que compartilham estados sejam reutilizados em vez de criados novamente.



Elementos do UML

- **Flyweight:** Interface para o ConcreteFlyweight e o UnsharedConcreteFlyweight. Pode ou não ser usado no python;
- **ConcreteFlyweight:** Implementa característica compartilhadas;
- **UnsharedConcreteFlyweight:** Pode usar as características do Concrete e contar características que não são compartilhadas;
- **FlyweightFactory:** Gerencia a criação dos objetos, permitindo a reutilização de objetos já criados.

Curiosidade

- O Flyweight foi proposto e usado na construção de um editor de texto chamado **Doc**
- Paul R. Calder and Mark A. Linton. The object-oriented implementation of a document editor. In Object-Oriented Programming Systems, Languages, and Applications Conference Proceedings, pages 154-165, Vancouver, British Columbia, Canada, October 1992. ACM Press.