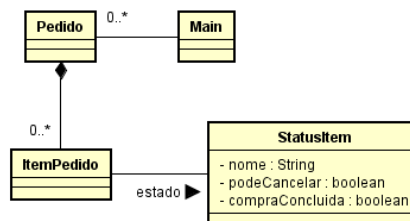


Exercício padrão Flyweight

O modelo abaixo exibe um diagrama de um sistema de vendas. A classe StatusItem pode ser compartilhada entre os itens do pedido(ItemPedido), pois ele é imutável, ou seja, seus atributos não mudam depois da sua criação. Ela pode ser criada com as seguintes configurações de atributos:

```
new StatusItem(Estado.CARRINHO, true, false);
new StatusItem(Estado.FECHADO, true, false);
new StatusItem(Estado.PAGO, true, true);
new StatusItem(Estado.ENVIADO, false, true);
new StatusItem(Estado.ENTREGUE, false, true);
```



Implemente o modelo primeiramente sem utilizar o padrão FlyWeight e depois implemente com o padrão. Crie duas classes Main uma utilizando o padrão e outra sem utilizar o padrão. Informe dentro do projeto em um txt qual a diferença de consumo de memória para as duas implementações.

Para a Main sem o padrão faça:

```
public static void main(String[] args) {
    ArrayList<Pedido> pedidos = new ArrayList();
    for(int i=0; i< 200000; i++){
        Pedido pedido = new Pedido();
        Item item1 = new Item();
        item1.setDescricao("lapis de cor modelo xyz");
        item1.setNome("lapis");
        item1.setStatus(new StatusItem(StatusItem.Estado.CARRINHO , true, false));
        Item item2 = new Item();
        item2.setDescricao("caneta cor azul");
        item2.setNome("caneta");
        item2.setStatus(new StatusItem(StatusItem.Estado.CARRINHO , true, false));
        Item item3 = new Item();
        item3.setDescricao("caneta cor vermelha");
        item3.setNome("caneta");
        item3.setStatus(new StatusItem(StatusItem.Estado.CARRINHO , true, false));
        pedido.addItemPedido(item1);
        pedido.addItemPedido(item2);
        pedido.addItemPedido(item3);
        pedidos.add(pedido);
    }
    System.out.println("KB: " + (double) (Runtime.getRuntime().totalMemory() - Runtime.getRuntime().freeMemory()) / 1024);
    System.out.println(pedidos.size());
}
```

Para a Main utilizando o padrão faça:

```
public static void main(String[] args) {
    ArrayList<Pedido> pedidos = new ArrayList();
    for(int i=0; i< 200000; i++){
        Pedido pedido = new Pedido();
        Item item1 = new Item();
        item1.setDescricao("lapis de cor modelo xyz");
        item1.setNome("lapis");
        item1.setStatus(FlyweightStatusItem.get(StatusItem.Estado.CARRINHO));
        Item item2 = new Item();
        item2.setDescricao("caneta cor azul");
        item2.setNome("caneta");
        item2.setStatus(FlyweightStatusItem.get(StatusItem.Estado.CARRINHO));
        Item item3 = new Item();
        item3.setDescricao("caneta cor vermelha");
        item3.setNome("caneta");
        item3.setStatus(FlyweightStatusItem.get(StatusItem.Estado.CARRINHO));
    }
}
```

```

        pedido.addItemPedido(item1);
        pedido.addItemPedido(item2);
        pedido.addItemPedido(item3);

        pedidos.add(pedido);
    }
    System.out.println("KB: " + (double) (Runtime.getRuntime().totalMemory() - Runtime.getRuntime().freeMemory()) / 1024);
    System.out.println(pedidos.size());
}

```

```
//classe StatusPedido
```

```

public class StatusItem {

    public enum Estado {CARRINHO, FECHADO, ENVIADO, ENTREGUE, PAGO};

    private Estado nome;

    private boolean podeCancelar;

    private boolean compraConcluida;

    public StatusItem(Estado nome, boolean podeCancelar, boolean compraConcluida){

        this.setNome(nome);

        this.podeCancelar = podeCancelar;

        this.compraConcluida = compraConcluida;

    }

    public Estado getNome() {

        return nome;

    }

    public void setNome(Estado nome) {

        this.nome = nome;

    }

    public boolean isPodeCancelar() {

        return podeCancelar;

    }

    public void setPodeCancelar(boolean podeCancelar) {

        this.podeCancelar = podeCancelar;

    }

    public boolean isCompraConcluida() {

        return compraConcluida;

    }

    public void setCompraConcluida(boolean compraConcluida) {

        this.compraConcluida = compraConcluida;

    }

}

```