

模版·动态规划(完全背包问题)

```
public class UnboundedKnapsack{
    public static int knapsack(Item[] items,int capacity){
        int[] dp=new int[capacity+1];

        for(int j=0;j<=capacity;j++){
            dp[j]=0;
        }

        for(Item item:items){
            for(int j=capacity;j>=item.weight;j--){
                for(int k=1;k<=j/item.weight;k++){
                    dp[j]=Math.max(dp[j],dp[j-k*item.weight]+k*item.value);
                }
            }
        }

        return dp[capacity];
    }

    public static class Item{
        int weight;
        int value;

        public Item(int weight,int value){
            this.weight=weight;
            this.value=value;
        }
    }

    public static void main(String[] args){
```

```
Item[] items={
    new Item(2,3),
    new Item(3,4),
    new Item(4,5),
    new Item(5,6)
};

int capacity=5;

int maxValue=knapsack(items,capacity);
System.out.println("可以装进背包的最大价值为:"+maxValue);
}
}
```