**Effects on User Portfolio When Buying Stocks**

When a user **buys stocks**, the following changes occur in their portfolio:

1. **quantity Increases:**
   * The number of shares the user owns is increased by the number of shares purchased (quantity).
   * Code:

python

Copy code

portfolio.quantity += quantity

1. **total\_investment Increases:**
   * The total investment reflects the total cost of all shares owned by the user.
   * The cost of the current trade (quantity × price) is added to total\_investment.
   * Code:

python

Copy code

portfolio.total\_investment += quantity \* price

1. **average\_purchase\_price Recalculated:**
   * The average purchase price is recalculated after the trade: average\_purchase\_price=total\_investmentquantity\text{average\\_purchase\\_price} = \frac{\text{total\\_investment}}{\text{quantity}}average\_purchase\_price=quantitytotal\_investment​
   * Code:

python

Copy code

if portfolio.quantity > 0:

portfolio.average\_purchase\_price = portfolio.total\_investment / portfolio.quantity

**Example (Buy Stock):**

* **Before Trade:**
  + quantity = 100, average\_purchase\_price = 50.00, total\_investment = 5,000.00
* **Trade Details:**
  + Buy 50 shares at a price of **60.00**.
* **After Trade:**
  + quantity = 150 (100 + 50)
  + total\_investment = 8,000.00 (5,000.00 + 50 × 60.00)
  + average\_purchase\_price = 53.33 (8,000.00 ÷ 150)

**Effects on User Portfolio When Selling Stocks**

When a user **sells stocks**, the following changes occur in their portfolio:

1. **quantity Decreases:**
   * The number of shares the user owns is reduced by the number of shares sold (quantity).
   * Code:

python

Copy code

portfolio.quantity -= quantity

1. **total\_investment Decreases:**
   * The total investment is reduced by the value of the shares sold (quantity × price).
   * Code:

python

Copy code

portfolio.total\_investment -= quantity \* price

1. **average\_purchase\_price Recalculated:**
   * If shares remain in the portfolio after the sale, the average\_purchase\_price is recalculated: average\_purchase\_price=total\_investmentquantity\text{average\\_purchase\\_price} = \frac{\text{total\\_investment}}{\text{quantity}}average\_purchase\_price=quantitytotal\_investment​
   * If no shares remain (quantity = 0), the average\_purchase\_price is reset to **0.00**.
   * Code:

python

Copy code

if portfolio.quantity > 0:

portfolio.average\_purchase\_price = portfolio.total\_investment / portfolio.quantity

else:

portfolio.average\_purchase\_price = Decimal('0.00')

**Example (Sell Stock):**

* **Before Trade:**
  + quantity = 100, average\_purchase\_price = 50.00, total\_investment = 5,000.00
* **Trade Details:**
  + Sell 50 shares at a price of **60.00**.
* **After Trade:**
  + quantity = 50 (100 - 50)
  + total\_investment = 2,000.00 (5,000.00 - 50 × 60.00)
  + average\_purchase\_price = 40.00 (2,000.00 ÷ 50)

**Key Notes on Effects**

1. **Buy Stocks:**
   * Increases quantity and total\_investment.
   * Adjusts average\_purchase\_price based on the total shares owned.
2. **Sell Stocks:**
   * Decreases quantity and total\_investment.
   * Adjusts average\_purchase\_price if shares remain; resets to 0.00 if no shares remain.
3. **Preventing Negative Quantity:**
   * The system should ensure the seller has enough shares to sell (quantity >= sell\_order.quantity).
   * Add a check in the \_handle\_sell\_order method to prevent negative quantities.

**Errors to Avoid**

1. **Negative Quantity in Portfolio:**
   * If the seller is allowed to sell more shares than they own, the quantity will become negative.
2. **Incorrect total\_investment Updates:**
   * If proceeds from sales are subtracted incorrectly, the total\_investment might show negative values.

**Final Portfolio Update Process**

Here’s the corrected process for updating portfolios:

**For Buyers:**

python

Copy code

portfolio.quantity += quantity

portfolio.total\_investment += quantity \* price

if portfolio.quantity > 0:

portfolio.average\_purchase\_price = portfolio.total\_investment / portfolio.quantity

portfolio.save()

**For Sellers:**

python

Copy code

if portfolio.quantity >= quantity: # Ensure the seller has enough shares

portfolio.quantity -= quantity

portfolio.total\_investment -= quantity \* price

if portfolio.quantity > 0:

portfolio.average\_purchase\_price = portfolio.total\_investment / portfolio.quantity

else:

portfolio.average\_purchase\_price = Decimal('0.00')

portfolio.save()

else:

raise ValidationError("Not enough shares to sell.")

Let me know if you’d like further assistance or clarification!