

Code:

```
public interface Tradeable1 {
    void buy(int quantity);
    void sell(int quantity);
}

class Stock implements Tradeable1 {
    private int AvailableQuantity;

    public Stock(int InitialQuantity) {
        this.AvailableQuantity = InitialQuantity;
    }

    @Override
    public void buy(int quantity) {
        if (quantity > 0) {
            AvailableQuantity += quantity;
            System.out.println(quantity + " Stocks are Bought!!");
            System.out.println("Available Quantity after updating is: " +
AvailableQuantity);
        } else {
            System.out.println("Invalid Quantity");
        }
    }

    @Override
    public void sell(int quantity) {
        if (quantity > 0 && quantity <= AvailableQuantity) {
            AvailableQuantity -= quantity;
            System.out.println(quantity + " Stocks are Sold!!");
            System.out.println("Available Quantity after updating is: " +
AvailableQuantity);
        } else {
            System.out.println("Invalid Quantity");
        }
    }

    public static void main(String[] args) {
        Stock ICDstock = new Stock(500);
    }
}
```

```

        ICDstock.buy(20);
        ICDstock.sell(10);
        ICDstock.buy(200);
        ICDstock.sell(600);
    }
}

```

Output:

```

// Tradeable1.java
import java.util.*;

public class Tradeable1 {
    public static void main(String[] args) {
        System.out.println("Invalid Quantity");
    }

    @Override
    public void sell(int quantity) {
        if (quantity > 0 && quantity <= AvailableQuantity) {
            AvailableQuantity -= quantity;
            System.out.println(quantity + " Stocks are Sold!!");
            System.out.println("Available Quantity after updating is: " + AvailableQuantity);
        } else {
            System.out.println("Invalid Quantity");
        }
    }

    public static void main(String[] args) {
        Stock ICDstock = new Stock(initialQuantity:500);

        ICDstock.buy(quantity:20);
        ICDstock.sell(quantity:10);
        ICDstock.buy(quantity:200);
        ICDstock.sell(quantity:600);
    }
}

```

```

PS D:\college docs\CS261 OOPS Lab\Lab 10 202211081> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
'C:\Users\shrey\AppData\Roaming\Code\User\workspaceStorage\7fb9946237cdc67776155a6dbf97081\redhat.java\jdk_ws\Lab 10 202211081_e458e607\bin' 'Stock'
20 Stocks are Bought!!
Available Quantity after updating is: 520
10 Stocks are Sold!!
Available Quantity after updating is: 510
200 Stocks are Bought!!
Available Quantity after updating is: 710
600 Stocks are Sold!!
Available Quantity after updating is: 110
PS D:\college docs\CS261 OOPS Lab\Lab 10 202211081>

```

Code:

```

public interface Tradeable2 {
    void buy(int quantity);
    void sell(int quantity);
}

class Stock implements Tradeable2 {
    private int availableQuantity;

    public Stock(int initialQuantity) {
        this.availableQuantity = initialQuantity;
    }
}

```

```
}

@Override
public void buy(int quantity) {
    if (quantity > 0) {
        availableQuantity += quantity;
        System.out.println(quantity + " Stocks are Bought!!");
        System.out.println("Available Quantity after updating is: " +
availableQuantity);
    } else {
        System.out.println("Invalid Quantity");
    }
}

@Override
public void sell(int quantity) {
    if (quantity > 0 && quantity <= availableQuantity) {
        availableQuantity -= quantity;
        System.out.println(quantity + " Stocks are Sold!!");
        System.out.println("Available Quantity after updating is: " +
availableQuantity);
    } else {
        System.out.println("Invalid Quantity");
    }
}

public static void main(String[] args) {
    Stock ICDstock = new Stock(1000);

    BuyThread buyThread1 = new BuyThread(ICDstock, 20);
    SellThread sellThread1 = new SellThread(ICDstock, 10);
    BuyThread buyThread2 = new BuyThread(ICDstock, 50);
    SellThread sellThread2 = new SellThread(ICDstock, 60);

    buyThread1.start();
    sellThread1.start();
    buyThread2.start();
    sellThread2.start();
}
}
```

```
class BuyThread extends Thread {
    private Stock stock;
    private int quantity;

    public BuyThread(Stock stock, int quantity) {
        this.stock = stock;
        this.quantity = quantity;
    }

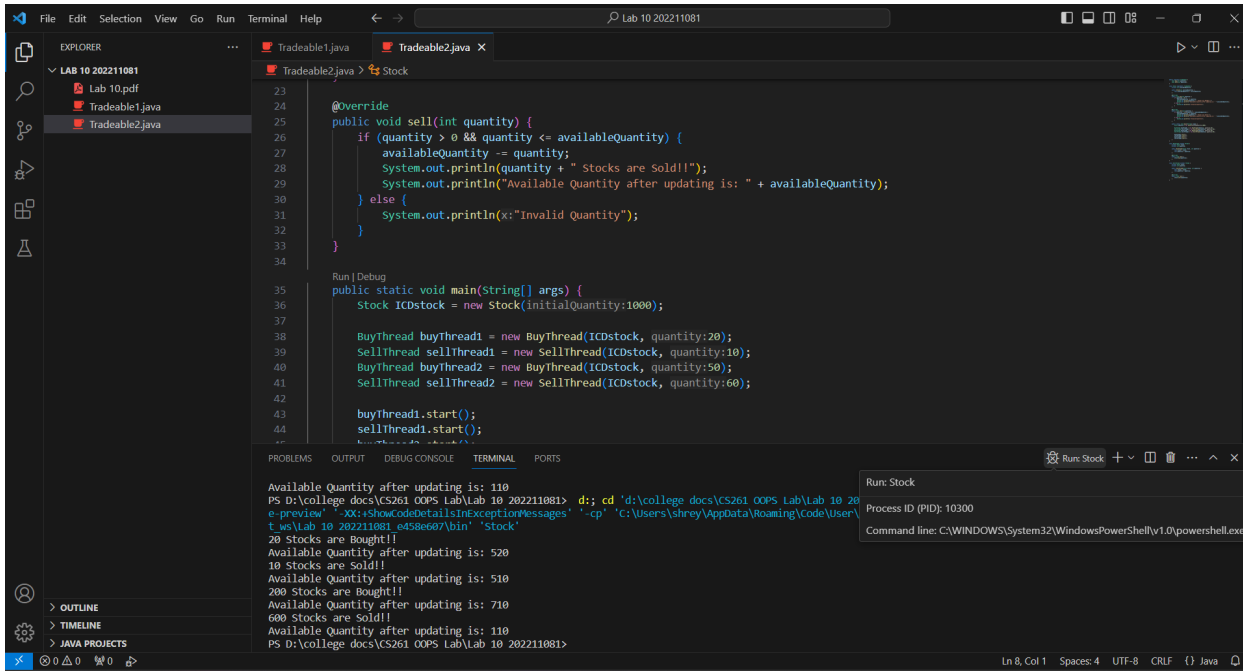
    @Override
    public void run() {
        stock.buy(quantity);
    }
}

class SellThread extends Thread {
    private Stock stock;
    private int quantity;

    public SellThread(Stock stock, int quantity) {
        this.stock = stock;
        this.quantity = quantity;
    }

    @Override
    public void run() {
        stock.sell(quantity);
    }
}
```

Output:



The screenshot displays an IDE with the following components:

- EXPLORER:** Shows a project named "LAB 10 202211081" containing "Lab 10.pdf", "Tradeable1.java", and "Tradeable2.java".
- Editor:** Displays the code for "Tradeable2.java". The code defines a `Stock` class with an `availableQuantity` attribute and a `sell` method. It also includes a `main` method that creates a `Stock` object and starts two threads, `buyThread1` and `sellThread1`, to simulate trading.
- TERMINAL:** Shows the output of the program. The output indicates that the initial quantity is 1000, and the threads successfully execute, updating the available quantity and printing the status of the stock.

```
23
24
25 @Override
26 public void sell(int quantity) {
27     if (quantity > 0 && quantity <= availableQuantity) {
28         availableQuantity -= quantity;
29         System.out.println(quantity + " Stocks are Sold!!");
30         System.out.println("Available Quantity after updating is: " + availableQuantity);
31     } else {
32         System.out.println("Invalid Quantity");
33     }
34 }
35
36 Run | Debug
37 public static void main(String[] args) {
38     Stock ICDstock = new Stock(initialQuantity:1000);
39
40     BuyThread buyThread1 = new BuyThread(ICDstock, quantity:20);
41     SellThread sellThread1 = new SellThread(ICDstock, quantity:10);
42     BuyThread buyThread2 = new BuyThread(ICDstock, quantity:50);
43     SellThread sellThread2 = new SellThread(ICDstock, quantity:60);
44
45     buyThread1.start();
46     sellThread1.start();
47 }
```

Available Quantity after updating is: 110  
PS D:\college docs\CS261 OOPS Lab\Lab 10 202211081> d; cd 'd:\college docs\CS261 OOPS Lab\Lab 10 202211081'; java -cp 'C:\Users\shrey\AppData\Roaming\Code\User\cache\lab 10 202211081\bin' 'Stock'

20 Stocks are Bought!!  
Available Quantity after updating is: 520  
10 Stocks are Sold!!  
Available Quantity after updating is: 510  
200 Stocks are Bought!!  
Available Quantity after updating is: 710  
600 Stocks are Sold!!  
Available Quantity after updating is: 110  
PS D:\college docs\CS261 OOPS Lab\Lab 10 202211081>

Run: Stock  
Process ID (PID): 10300  
Command line: C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe