```java
package edu.cpp.cs5800.VendingMachine;

public class Driver {
    public static void main(String[] args) {
        VendingMachine vendingMachine = new VendingMachine();
        vendingMachine.printAvailableSnacks();

        vendingMachine.selectSnack(6);
        vendingMachine.insertMoney(6.5);
        vendingMachine.dispenseSnack();
        System.out.println();

        vendingMachine.printAvailableSnacks();
        vendingMachine.selectSnack(6);
        vendingMachine.insertMoney(5);
        vendingMachine.dispenseSnack();
        System.out.println();

        vendingMachine.selectSnack(3);
        vendingMachine.insertMoney(2);
        vendingMachine.dispenseSnack();
        System.out.println();

        vendingMachine.selectSnack(4);
        vendingMachine.insertMoney(3.5);
        vendingMachine.dispenseSnack();
        System.out.println();

    }
}

```

```java
package edu.cpp.cs5800.VendingMachine;

import edu.cpp.cs5800.VendingMachine.snacks.*;
import edu.cpp.cs5800.VendingMachine.states.Idle;
import edu.cpp.cs5800.VendingMachine.states.StateOfVendingMachine;

import java.util.HashMap;
import java.util.Map;

public class VendingMachine {
    private StateOfVendingMachine state = new Idle(this);
    private int selectedSnack;
    private double insertedAmount;
    private SnackDispenseHandler snackDispenseHandler;

    private Map<Integer, Snack> stocks = new HashMap<>();

    public VendingMachine() {
        Snack.resetId();
        Coke coke = new Coke(15, 3);
        Pepsi pepsi = new Pepsi(15, 3);
        Cheetos cheetos = new Cheetos(15, 2.5);
        Doritos doritos = new Doritos(15, 3.5);
        KitKat kitkat = new KitKat(2, 5);
        Snickers snickers = new Snickers(1, 4);

        SnackDispenseHandler snickersHandler = new SnackDispenseHandler(snickers, null);
        SnackDispenseHandler kitkatHandler = new SnackDispenseHandler(kitkat, snickersHandler);
        SnackDispenseHandler doritosHandler = new SnackDispenseHandler(doritos, kitkatHandler);
        SnackDispenseHandler cheetosHandler = new SnackDispenseHandler(cheetos, doritosHandler);
        SnackDispenseHandler pepsiHandler = new SnackDispenseHandler(pepsi, cheetosHandler);
        snackDispenseHandler = new SnackDispenseHandler(coke, pepsiHandler);

        stocks.put(coke.getId(), coke);
        stocks.put(pepsi.getId(), pepsi);
        stocks.put(cheetos.getId(), cheetos);
        stocks.put(doritos.getId(), doritos);
        stocks.put(kitkat.getId(), kitkat);
        stocks.put(snickers.getId(), snickers);
    }

    public void printAvailableSnacks() {
        for (int key: stocks.keySet().stream().sorted().toList()) {
            Snack snack = stocks.get(key);
            System.out.println(key + ": " + snack.getName() + ", price: $" + snack.getPrice() + ", stock: " + snack.getQuantity());
        }
        System.out.println();
    }
```

```
49
50      public SnackDispenseHandler getSnackDispenseHandler() {
51          return snackDispenseHandler;
52      }
53
54      public Snack getSnack(int number) {
55          return stocks.getOrDefault(number, null);
56      }
57
58      public void setState(StateOfVendingMachine state) {
59          System.out.println("State transition: " + this.state + "-->" + state);
60          System.out.println("--------------------");
61          this.state = state;
62      }
63
64      public StateOfVendingMachine getState() {
65          return state;
66      }
67
68      public int getSelectedSnack() {
69          return selectedSnack;
70      }
71
72      public void setSelectedSnack(int selectedSnack) {
73          this.selectedSnack = selectedSnack;
74      }
75
76      public double getInsertedAmount() {
77          return insertedAmount;
78      }
79
80      public void setInsertedAmount(double insertedAmount) {
81          this.insertedAmount = insertedAmount;
82      }
83
84      public void selectSnack(int number) {
85          this.state.selectSnack(number);
86      }
87
88      public void insertMoney(double amount) {
89          this.state.insertMoney(amount);
90      }
91
92      public void dispenseSnack() {
93          this.state.dispenseSnack();
94      }
95  }
96
```

```java
package edu.cpp.cs5800.VendingMachine;

import edu.cpp.cs5800.VendingMachine.snacks.Snack;

public class SnackDispenseHandler {
    private Snack snack;
    private SnackDispenseHandler nextHandler;

    public SnackDispenseHandler(Snack snack, SnackDispenseHandler nextHandler) {
        this.snack = snack;
        this.nextHandler = nextHandler;
    }

    public boolean dispense(int id, double amount) {
        System.out.print(this.snack.getName() + "Handler");
        if (this.snack.getId() == id) {
            System.out.println();
            if (this.snack.getQuantity() == 0) {
                System.out.println("Not enough quantity for item: " + this.snack.getName
());
                System.out.printf("Returning money inserted: $%2.2f\n", amount);
            } else if (this.snack.getPrice() <= amount) {
                System.out.println("Dispensing: " + this.snack.dispense() + ", price: $" +
this.snack.getPrice());
                if (this.snack.getPrice() < amount) {
                    System.out.printf("Returning Change: $%2.2f\n", amount - this.snack.
getPrice());
                }
                return true;
            } else {
                System.out.println("Money inserted is not enough: " + this.snack.getName
() + ", price: $" + this.snack.getPrice());
            }
        } else if (this.nextHandler != null) {
            System.out.print("->");
            return this.nextHandler.dispense(id, amount);
        } else {
            System.out.println("Item id not found: " + id);
        }

        return false;
    }
}

```

```java
package edu.cpp.cs5800.VendingMachine.snacks;

public class Coke extends Snack {
    public Coke(int quantity, double price) {
        super("Coke", quantity, price);
    }

}

```

```java
package edu.cpp.cs5800.VendingMachine.snacks;

public class Pepsi extends Snack {

    public Pepsi(int quantity, double price) {
        super("Pepsi", quantity, price);
    }
}

```

```java
package edu.cpp.cs5800.VendingMachine.snacks;

public abstract class Snack {
    private static int idIncremnt = 1;
    private int id;
    private String name;
    private int quantity;
    private double price;

    public Snack(String name, int quantity, double price) {
        this.id = idIncremnt++;
        this.name = name;
        this.quantity = quantity;
        this.price = price;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public int getQuantity() {
        return quantity;
    }

    public double getPrice() {
        return price;
    }

    public String dispense() {
        quantity--;
        return name;
    }

    public static void resetId() {
        idIncremnt = 1;
    }

    @Override
    public String toString() {
        return "Snack{" +
                "name='" + name + '\'' +
                ", quantity=" + quantity +
                ", price=" + price +
                '}';
    }
}

```

```java
package edu.cpp.cs5800.VendingMachine.snacks;

public class KitKat extends Snack {

    public KitKat(int quantity, double price) {
        super("KitKat", quantity, price);
    }
}

```

```java
package edu.cpp.cs5800.VendingMachine.snacks;

public class Cheetos extends Snack {

    public Cheetos(int quantity, double price) {
        super("Cheetos", quantity, price);
    }
}

```

```java
package edu.cpp.cs5800.VendingMachine.snacks;

public class Doritos extends Snack {

    public Doritos(int quantity, double price) {
        super("Doritos", quantity, price);
    }
}

```

```java
package edu.cpp.cs5800.VendingMachine.snacks;

public class Snickers extends Snack {

    public Snickers(int quantity, double price) {
        super("Snickers", quantity, price);
    }
}

```

```java
package edu.cpp.cs5800.VendingMachine.states;

import edu.cpp.cs5800.VendingMachine.VendingMachine;
import edu.cpp.cs5800.VendingMachine.snacks.Snack;

public class Idle extends StateOfVendingMachine {

    public Idle(VendingMachine vendingMachine) {
        super(vendingMachine);
    }

    @Override
    public String selectSnack(int number) {
        Snack snack = this.vendingMachine.getSnack(number);
        String message = "You have selected: " + number + " (" + snack.getName() + ")";
        System.out.println(message);
        this.vendingMachine.setSelectedSnack(number);
        this.vendingMachine.setState(new WaitingForMoney(this.vendingMachine));
        return message;
    }

    @Override
    public String insertMoney(double amount) {
        String message = "Invalid request: Please first select a snack!";
        System.out.println(message);
        return message;
    }

    @Override
    public String dispenseSnack() {
        String message = "Invalid request: Please first select a snack!";
        System.out.println("Invalid request: Please first select a snack!");
        return message;
    }

    public String toString() {
        return "Idle";
    }
}

```

```java
package edu.cpp.cs5800.VendingMachine.states;

import edu.cpp.cs5800.VendingMachine.VendingMachine;

public class DispensingSnack extends StateOfVendingMachine {

    public DispensingSnack(VendingMachine vendingMachine) {
        super(vendingMachine);
    }

    @Override
    public String selectSnack(int number) {
        String message = "Invalid request: Currently dispensing snack!";
        System.out.println("Invalid request: Currently dispensing snack!");
        return message;
    }

    @Override
    public String insertMoney(double amount) {
        String message = "Invalid request: Currently dispensing snack!";
        System.out.println("Invalid request: Currently dispensing snack!");
        return message;
    }

    @Override
    public String dispenseSnack() {
        int selectedItem = this.vendingMachine.getSelectedSnack();
        double amt = this.vendingMachine.getInsertedAmount();
        System.out.print("Chain of responsibility: ");
        this.vendingMachine.getSnackDispenseHandler().dispense(selectedItem, amt);
        String message = "Completed transaction!";
        System.out.println(message);
        this.vendingMachine.setState(new Idle(this.vendingMachine));
        return message;
    }

    public String toString() {
        return "DispensingSnack";
    }
}

```

```java
package edu.cpp.cs5800.VendingMachine.states;

import edu.cpp.cs5800.VendingMachine.VendingMachine;

public class WaitingForMoney extends StateOfVendingMachine {

    public WaitingForMoney(VendingMachine vendingMachine) {
        super(vendingMachine);
    }

    @Override
    public String selectSnack(int number) {
        String message = "Invalid request: Please inserted money!";
        System.out.println(message);
        return message;
    }

    @Override
    public String insertMoney(double amount) {
        String message = "You have inserted: $" + amount;
        System.out.println(message);
        this.vendingMachine.setInsertedAmount(amount);
        this.vendingMachine.setState(new DispensingSnack(this.vendingMachine));
        return message;
    }

    @Override
    public String dispenseSnack() {
        String message = "Invalid request: Please inserted money!";
        System.out.println("Invalid request: Please inserted money!");
        return message;
    }

    public String toString() {
        return "WaitingForMoney";
    }
}

```

```java
package edu.cpp.cs5800.VendingMachine.states;

import edu.cpp.cs5800.VendingMachine.VendingMachine;

public abstract class StateOfVendingMachine {
    protected VendingMachine vendingMachine;

    public StateOfVendingMachine(VendingMachine vendingMachine) {
        this.vendingMachine = vendingMachine;
    }

    public abstract String selectSnack(int number);

    public abstract String insertMoney(double amount);

    public abstract String dispenseSnack();
}

```