This is the stack implementation in JAVA

```
package ds;


public class BasicStack<X> {

    private X[] data;
    private int stackPointer;

    public BasicStack(){
        data = (X[]) new Object[1000];
        stackPointer = 0;
    }

    public void push(X item){
        data[stackPointer++]= item;
    }

    public X pop(){
        if(stackPointer == 0) {
            throw new IllegalStateException("No more items
on the stack");
        }
        return data[--stackPointer];
    }

    public boolean contains(X item){
        boolean found = false;

        for(int i=0;i< stackPointer; i++){
            if(data[i].equals(item)) {
                found = true;
                break;
            }
        }

        return found;
```

```
    }

    public X access(X item){
        while(stackPointer > 0){
            X topItem = pop();
            if(topItem.equals(item))
                return topItem;
        }

        // if we didn't find the item in the stack throw an
exception
        throw new IllegalArgumentException("Can't find your
item in the stack" + item);
    }

    public int size(){
        return stackPointer;
    }
}
```

As it is of Generic type the stack can be used with any data type. Here it is used in CardStackApp using String:

```
package app;

import ds.BasicStack;

public class CardStackApp {
    BasicStack<String> stack = new BasicStack<String>();

    public static void main(String[] args) {
        CardStackApp app = new CardStackApp();
        app.stackCards();
        app.unstackCards();

        //restack cards
        app.stackCards();
```

```
    //how many cards are on the deck
    app.desckSize();

    //do we have queen of hearts in the deck
    app.containsCard("Queen of Hearts");

    //do we have a joker
    app.containsCard("Joker");

    //go to the king of diamonds
    app.goToCard("King of Diamonds");

    //now how many cards are on the deck
    app.desckSize();
}

public void stackCards() {
    //stack the spade suit
    stack.push("Ace of Spades");
    stack.push("2 of Spades");
    stack.push("3 of Spades");
    stack.push("4 of Spades");
    stack.push("5 of Spades");
    stack.push("6 of Spades");
    stack.push("7 of Spades");
    stack.push("8 of Spades");
    stack.push("9 of Spades");
    stack.push("10 of Spades");
    stack.push("Jack of Spades");
    stack.push("Queen of Spades");
    stack.push("King of Spades");

    //stack the diamond suit
    stack.push("Ace of Diamonds");
    stack.push("2 of Diamonds");
    stack.push("3 of Diamonds");
    stack.push("4 of Diamonds");
    stack.push("5 of Diamonds");
```

```
stack.push("6 of Diamonds");
stack.push("7 of Diamonds");
stack.push("8 of Diamonds");
stack.push("9 of Diamonds");
stack.push("10 of Diamonds");
stack.push("Jack of Diamonds");
stack.push("Queen of Diamonds");
stack.push("King of Diamonds");

//stack the club suit
stack.push("Ace of Clubs");
stack.push("2 of Clubs");
stack.push("3 of Clubs");
stack.push("4 of Clubs");
stack.push("5 of Clubs");
stack.push("6 of Clubs");
stack.push("7 of Clubs");
stack.push("8 of Clubs");
stack.push("9 of Clubs");
stack.push("10 of Clubs");
stack.push("Jack of Clubs");
stack.push("Queen of Clubs");
stack.push("King of Clubs");

//stack the heart suit
stack.push("Ace of Hearts");
stack.push("2 of Hearts");
stack.push("3 of Hearts");
stack.push("4 of Hearts");
stack.push("5 of Hearts");
stack.push("6 of Hearts");
stack.push("7 of Hearts");
stack.push("8 of Hearts");
stack.push("9 of Hearts");
stack.push("10 of Hearts");
stack.push("Jack of Hearts");
stack.push("Queen of Hearts");
stack.push("King of Hearts");
```

```
    }

    public void unstackCards() {
        //now pull the cards off the stack and print them
        while(stack.size() > 0) {
            System.out.println(stack.pop());
        }
    }

    public void containsCard(String card) {
        System.out.println(stack.contains(card));
    }

    public void goToCard(String card) {
        System.out.println(stack.access(card));
    }

    public void desckSize() {
        System.out.println(stack.size());
    }
}
```