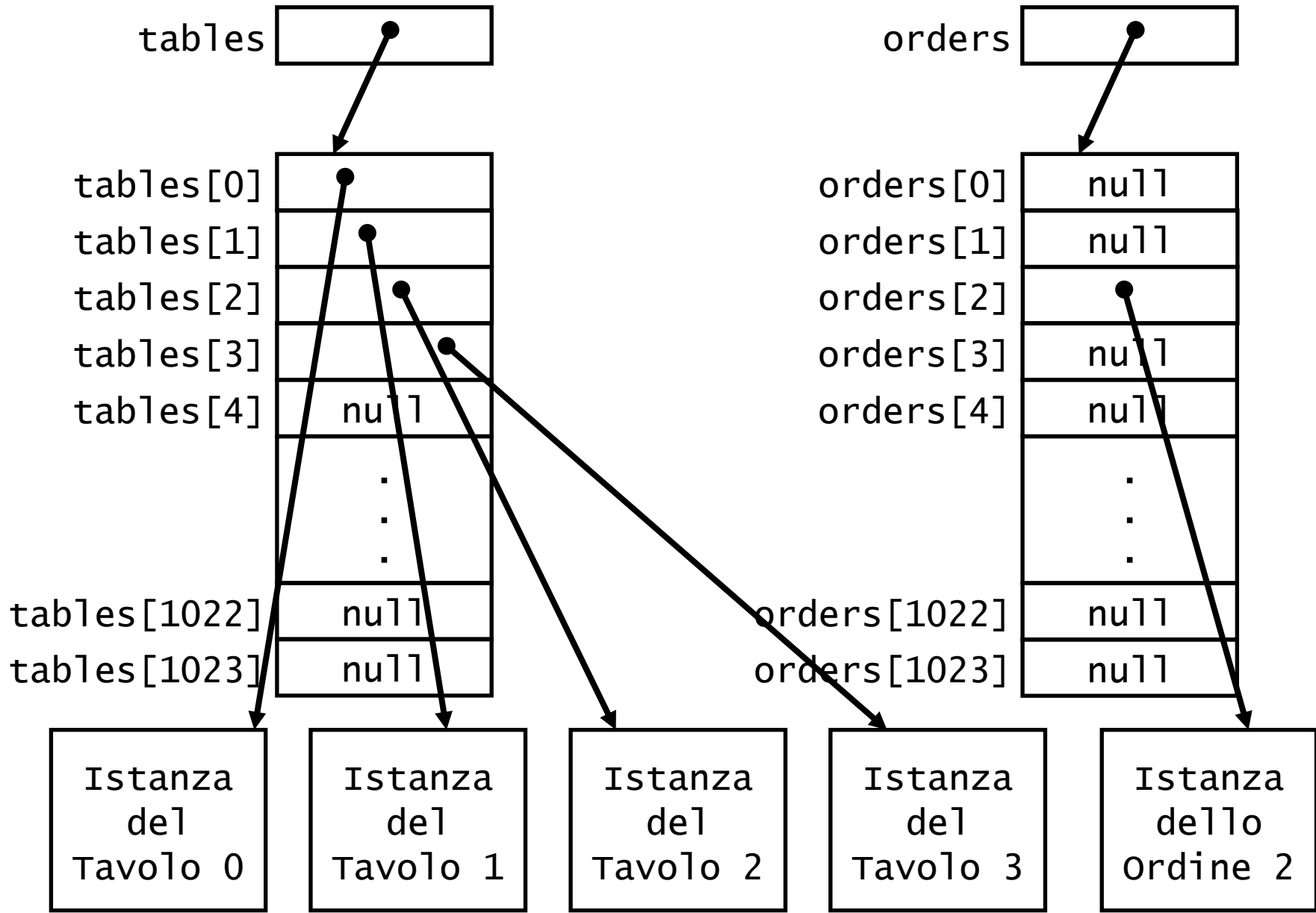




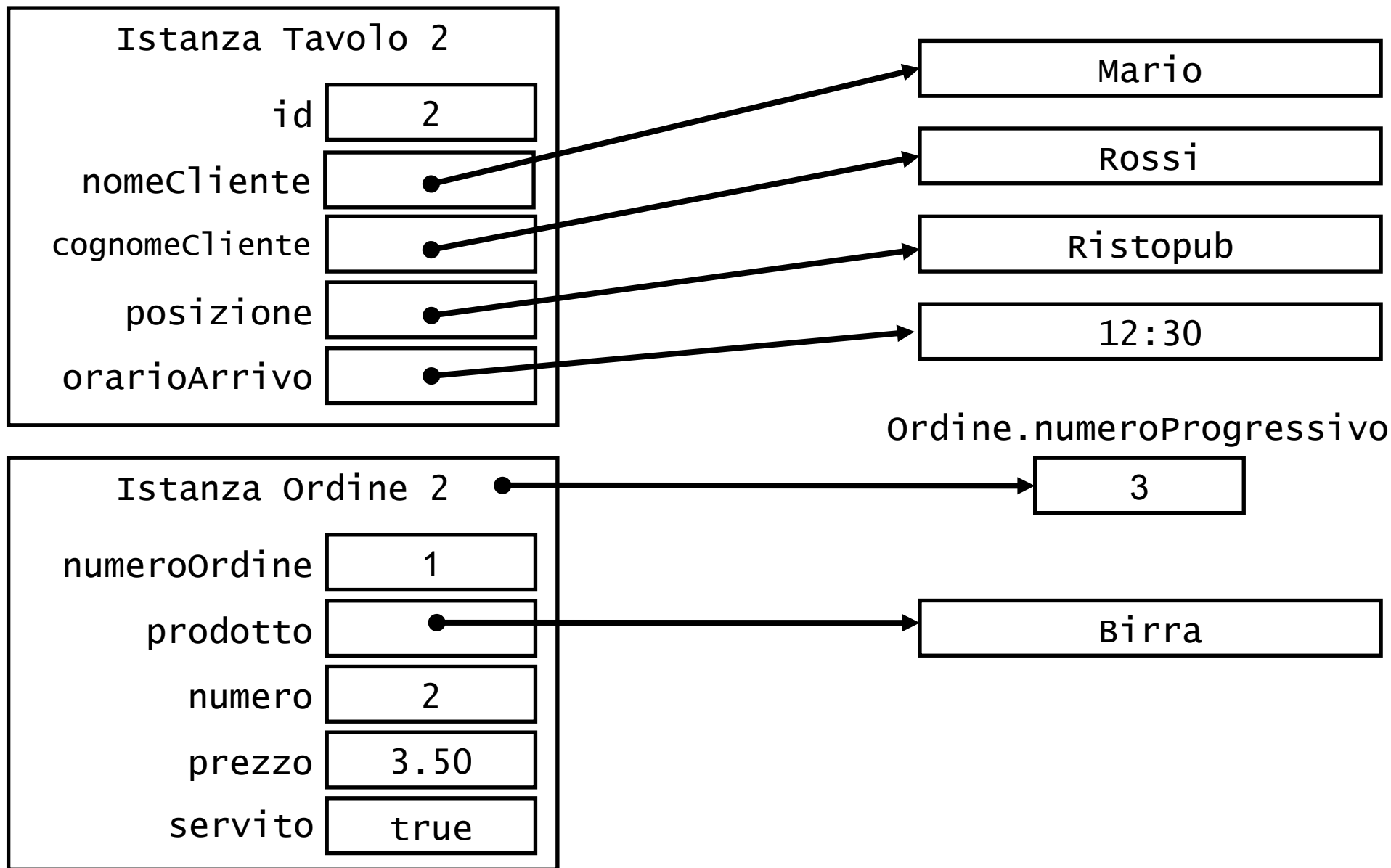
Appello del 13/06/2014

Soluzione passo-passo

Struttura Database



Struttura Istanze di Tavolo ed Ordine



Inizializzazione Database (E)

```
public class Appello_13062014 {  
    static final int MAX_ELEM = 1024;  
    public static void main(String[] args) {  
        Tavolo[] tables = new Tavolo[MAX_ELEM];  
        Ordine[] orders = new Ordine[MAX_ELEM];  
  
        tables[0] =  
            new Tavolo(0, "Mario", "Rossi", "Ristopub", "12:30");  
        tables[1] =  
            new Tavolo(1, "Pietro", "Rossi", "Pizzeria", "9:50");  
        tables[2] =  
            new Tavolo(0, "Mario", "Rossi", "Ristopub", "12:30");  
        tables[3] =  
            new Tavolo(2, "Gio'", "Verdi", "Veranda", "11:25");  
        tables[4] =  
            new Tavolo(2, "Gio'", "Verdi", "Veranda", "11:25");  
  
        orders[0] = new Ordine("Acqua", 1, 2, true);  
        orders[2] = new Ordine("Birra", 2, 3.50, true);  
        orders[3] = new Ordine("Vino", 1, 10, true);  
        orders[4] = new Ordine("Acqua", 1, 2, true);  
    }  
}
```

Stampa Database (E)

```
public static Appello_13062014 {  
  
    ...  
  
    public static int contaTavoli(Tavolo[] tavoli) {  
        int count = 0;  
        while (count < MAX_ELEM && tavoli[count] != null) {  
            count++;  
        }  
        return count;  
    }  
  
    public static void stampaDB(Tavolo[] tavoli, Ordine[] ordini) {  
        int n = contaTavoli(tavoli);  
        for (int i= 0; i<n; i++) {  
            if (ordini[i] != null) {  
                System.out.println(tavoli[i] + " - " + ordini[i]);  
            }  
            else {  
                System.out.println(tavoli[i]);  
            }  
        }  
    }  
  
    ...  
  
}
```

Stampa Ordini (A)

```
public static Appello_13062014 {  
  
    ...  
  
    public static int contaOrdini(Ordine[] ordini) {  
        int count = 0;  
        for (int i=0; i < MAX_ELEM; i++) {  
            if (ordini[i] != null) {  
                count++;  
            }  
        }  
        return count;  
    }  
  
    ...  
  
    public static void main(String[] args) {  
  
        ...  
  
        System.out.println("\nA.");  
        System.out.println("Il numero di ordini effettivi è " +  
                            contaOrdini(orders));  
    }  
  
}
```

Ordinamento Tavoli (B)

```
public static Appello_13062014 {  
  
    ...  
  
    public static int estraiOra(String orario) {  
        int idx = orario.indexOf(':', 0);  
        String ora = orario.substring(0, idx);  
        return Integer.parseInt(ora);  
    }  
  
    ...  
  
    public static void scambiaOrdine(Ordine[] v, int i, int j) {  
        Ordine tmp = v[i];  
        v[i] = v[j];  
        v[j] = tmp;  
    }  
  
    ...  
  
    public static void scambiaTavolo(Tavolo[] v, int i, int j) {  
        Tavolo tmp = v[i];  
        v[i] = v[j];  
        v[j] = tmp;  
    }  
  
    ...  
  
}
```

Ordinamento Tavoli (B)

```
public static Appello_13062014 {  
  
    ...  
  
    public static void ordinaArrivo(Tavolo[] tavoli, Ordine[] ordini) {  
        int n = contaTavoli(tavoli);  
        for (int i=0; i < n-1; i++) {  
            int min = i;  
            for (int j=i+1; j < n; j++) {  
                if (estraiOra(tavoli[j].orarioArrivo) <  
                    estraiOra(tavoli[min].orarioArrivo) {  
                    min = j;  
                }  
            }  
            scambiaTavolo(tavoli, i, min);  
            scambiaOrdine(ordini, i, min);  
        }  
    }  
  
    ...  
  
    public static void main(String[] args) {  
        ...  
        System.out.println("\nB.");  
        System.out.println("\nPrima dell'ordinamento:");  
        stampaDB(tables, orders);  
        System.out.println("\nDopo l'ordinamento:");  
        ordinaArrivo(tables, orders);  
        stampaDB(tables, orders);  
    }  
  
}
```


Conto Tavolo 0 (C)

```
public static Appello_13062014 {  
  
    ...  
  
    public static double conto(Tavolo[] tavoli, Ordine[] ordini, int id) {  
        int n = contaTavoli(tavoli);  
        double tot = 0;  
        for (int i=0; i < n; i++) {  
            if (tavoli[i].getId() == id) {  
                if (ordini[i] != null && ordini[i].servito) {  
                    tot += ordini[i].prezzo * ordini[i].numero;  
                }  
            }  
        }  
        return tot;  
    }  
  
    ...  
  
    public static void main(String[] args) {  
  
        ...  
  
        System.out.println("\nC.");  
        double totTav0 = conto(tavoli, ordini, 0);  
        System.out.println("\nIl Conto del Tavolo #0 è " + totTav0 + "€");  
    }  
  
}
```

Eliminazione Ordini (D)

```
public static Appello_13062014 {
    ...
    public static boolean eliminaOrdineOra(Tavolo[] tavoli,
                                           Ordine[] ordini, int ora) {

        int n = contaTavoli(tavoli);
        int ordDel = 0;
        for (int i = 0; i < n; i++) {
            if (estraiOra(tavoli[i].orarioArrivo) < ora) {
                tavoli[i] = null;
                ordini[i] = null;
                ordDel++;
            }
        }
        orders, 0
        if (ordDel > 0) {
            // bisogna compattare!!!
            for (int i=0; i < n-1; i++) {
                for (int j=i+1; j < n; j++) {
                    if (tavoli[i] == null && tavoli[j] != null) {
                        scambiaTavolo(tavoli, i, j);
                        scambiaOrdine(ordini, i, j);
                    }
                }
            }
            return true;
        }
        else
            return false;
    }
    ...
}
```

Eliminazione Ordini (D)

```
public static Appello_13062014 {  
  
    ...  
  
    public static void main(String[] args) {  
  
        ...  
  
        System.out.println("\nD.");  
        if (eliminaOrdineOre(tables, orders, 12)) {  
            System.out.println("\nGli ordini rimanenti sono:");  
            stampaDB(tables, orders);  
        }  
        else {  
            System.out.println("\nNessuna eliminazione.");  
        }  
    }  
}
```