



Un esempio riassuntivo

Esempio: Tombola!



Tombola!

| | | | | | | | | |
|---|----|----|----|----|----|----|----|----|
| | 17 | 22 | | 45 | | 66 | | 83 |
| 1 | 19 | | 30 | 48 | | | | 70 |
| | | 29 | 37 | | 53 | | 74 | 86 |
| | | 58 | 31 | | 23 | | 14 | 89 |

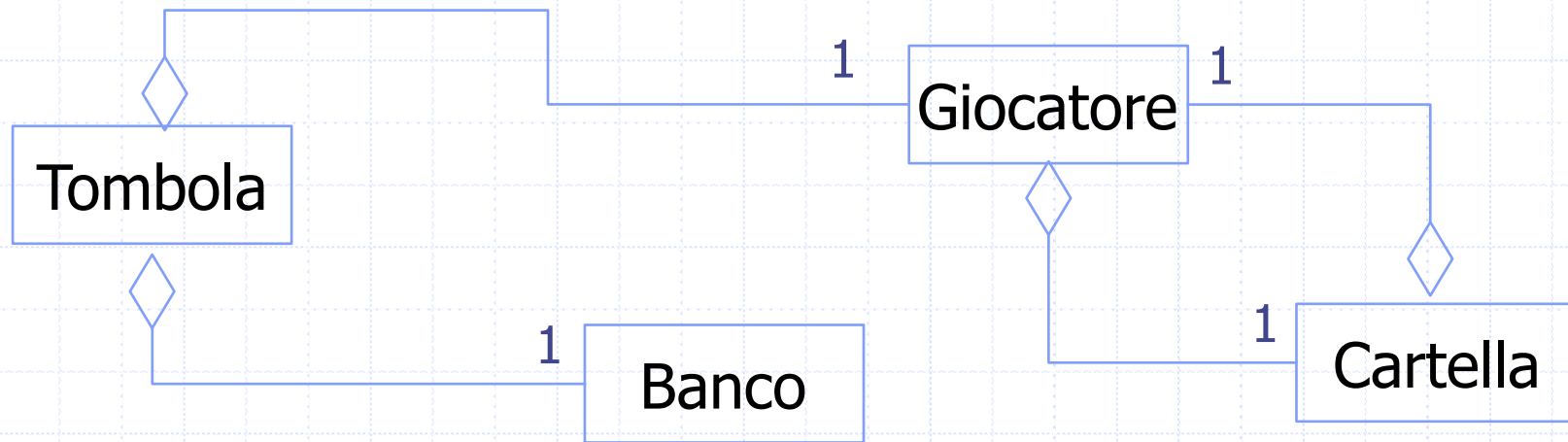
Tombola - esercizio

Esercizio:

Modificare il codice aggiungendo un numero arbitrario di giocatori, ciascuno con un numero arbitrario di cartelle

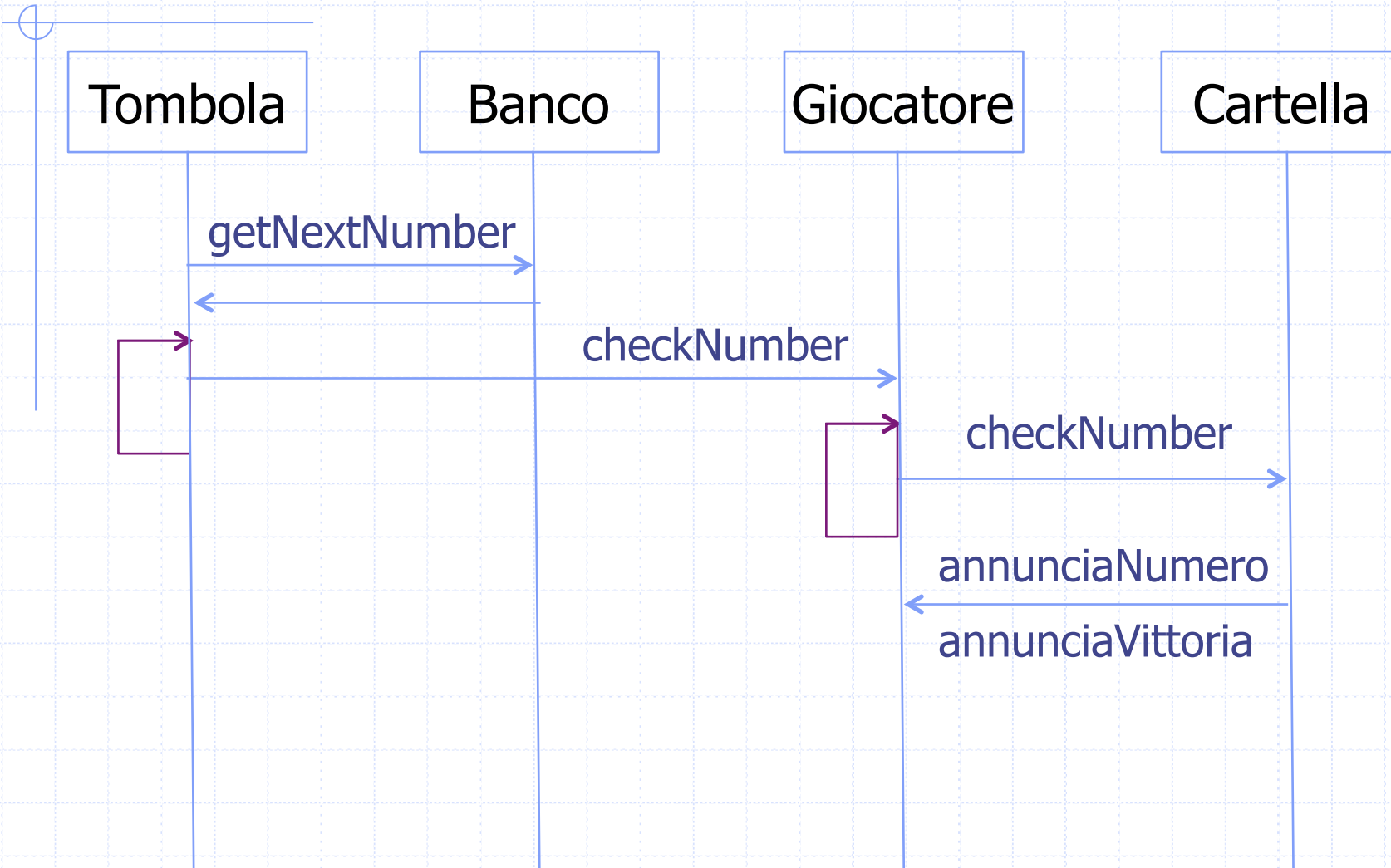
4

Class diagram



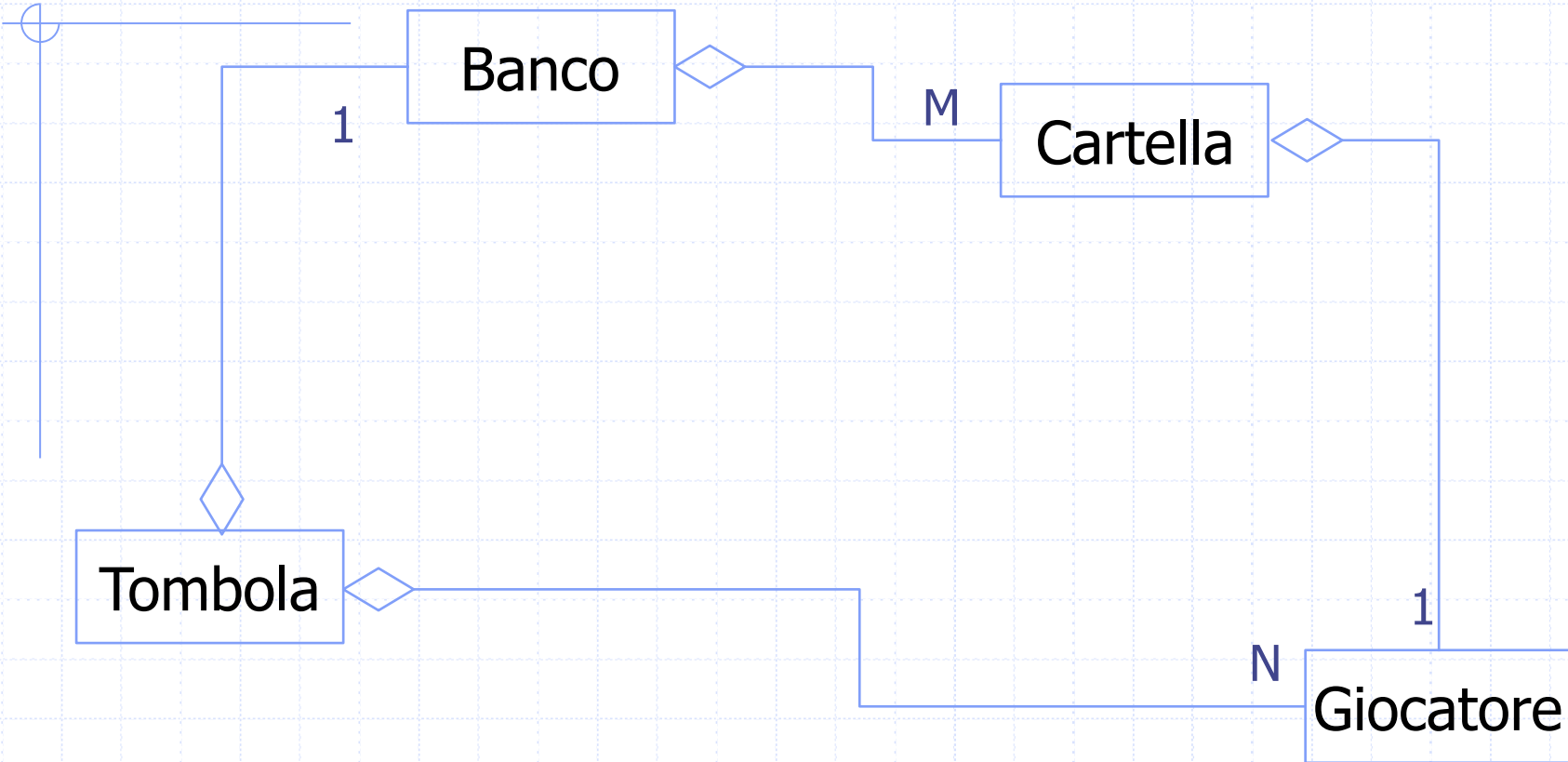
5

Activity diagram (swim lanes)



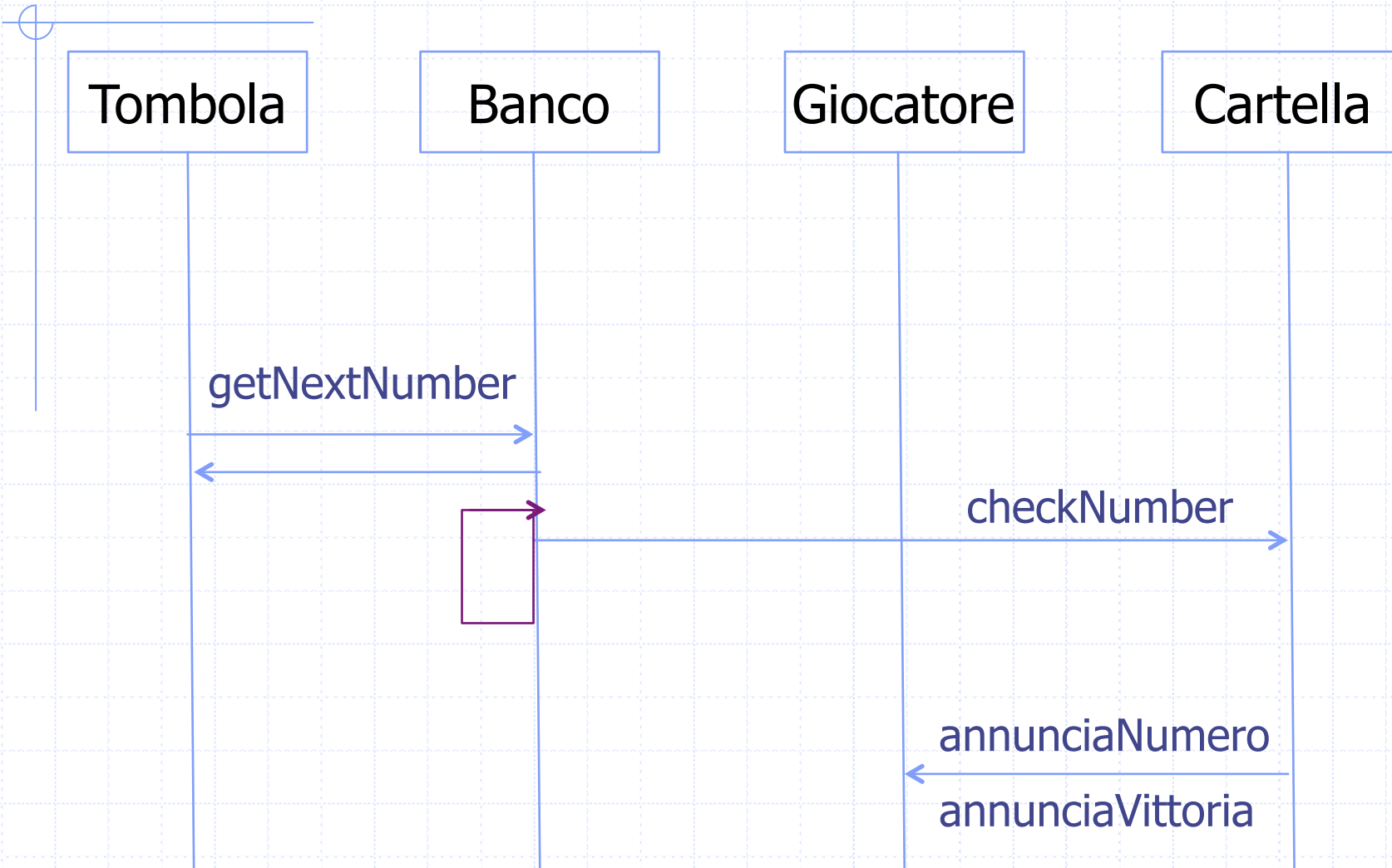
6

Class diagram



7

Activity diagram (swim lanes)



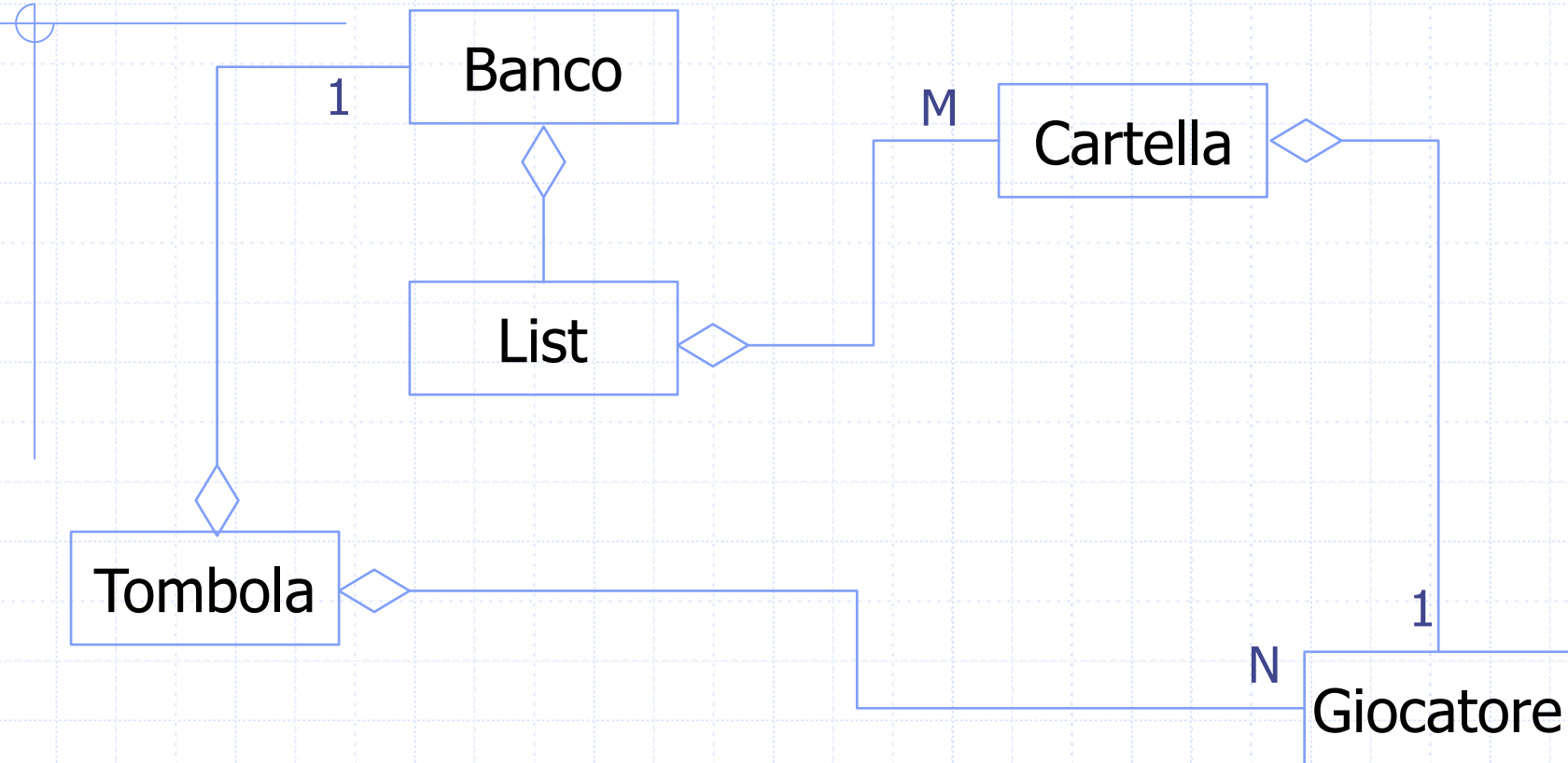
8

Listener

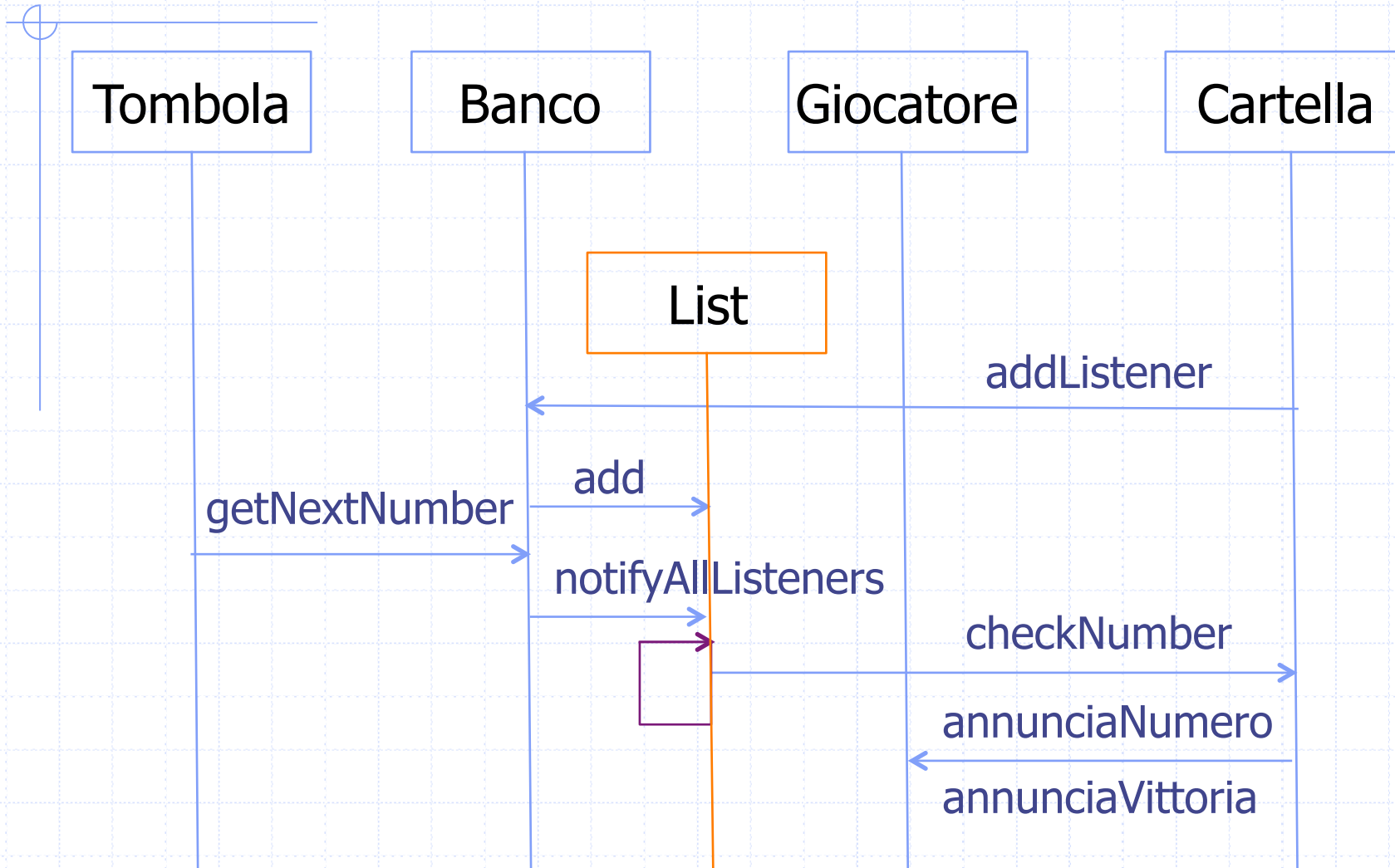
Introduciamo la nozione di "ascoltatore"

secondo il paradigma "publish & subscribe"

Class diagram



Activity diagram (swim lanes)



11

Common

```
package tombola;  
import java.util.Random;  
public class Common {  
    static final int NCELLS=3;  
    static final int MAXNUM=10;  
    static final Random generatore =  
        new Random(System.currentTimeMillis());  
}
```



invariato

Giocatore



Eliminiamo qualche pezzo

```
package tombola;
public class Giocatore {
    public String name;
    // private Cartella cartella; non serve
    Giocatore(String name){
        this.name=name;
        // cartella=new Cartella(this); non serve
    }
    void checkNumber(int x){
    cartella.checkNumber(x);
    }
}
```

Giocatore

```
void annunciaNumero(int num, int cartellaId){  
    System.out.println(name+" ha il numero  
"+num+" in cartella "+cartellaId);  
}
```

```
void annunciaVittoria(int cartellaId) {  
    System.out.println(name+" ha vinto con  
cartella "+cartellaId);
```

```
    cartella.printOriginale();
```

```
    System.exit(1);
```

```
}
```

```
}
```



Quasi invariato

Cartella

```
public class Cartella {
```

```
    private HashSet numeri = new HashSet();  
    private HashSet mancanti = new HashSet();  
    private Giocatore proprietario=null;  
    private int id=0;  
    static int nCartelle=0;
```

```
    Cartella(Giocatore g) {
```

```
        id=++nCartelle;
```

```
        proprietario=g;
```

```
        for (int i = 1; i <= Common.NCELLS; i++) {
```

```
            boolean creatoNuovoNumero = false;
```

```
            do {
```

```
                int x = Common.generatore.nextInt(Common.MAXNUM)+1;
```

```
                creatoNuovoNumero = numeri.add(new Integer(x));
```

```
                if (creatoNuovoNumero) System.out.println("aggiunto "+ x);
```

```
            } while (!creatoNuovoNumero);
```

```
        }
```

```
        mancanti.addAll(numeri);
```

```
    }
```

```
package tombola;
```

```
import java.util.HashSet;  
import java.util.Iterator;
```



Cartella

```
public boolean checkNumber(int x) {  
    boolean result = mancanti.remove(new Integer(x));  
    if(proprietario!=null) {  
        if (result) proprietario.annunciaNumero(x, id);  
        if (mancanti.isEmpty()) proprietario.annunciaVittoria(id);  
    }  
    return result;  
}
```

```
private void print(HashSet list) {  
    Iterator iter = list.iterator();  
    while (iter.hasNext()) {  
        System.out.print(iter.next()+" ");  
    }  
    System.out.println();  
}
```

```
public void printOriginale() {print(neri);}  
public void printCurrent() {print(mancanti);}  
}
```



invariato

Banco

```
package tombola;
```

```
import java.util.LinkedList;  
import java.util.List;
```

```
public class Banco {  
    List sacchetto;  
    List<Cartella> cartelle;  
  
    public Banco() {  
        cartelle= new LinkedList();  
        sacchetto= new LinkedList();  
        for (int i=1; i<=Common.MAXNUM;i++) {  
            sacchetto.add(new Integer(i));  
        }  
    }  
}
```


Banco – Lista di ascoltatori

```
void addListener(Cartella c){  
    cartelle.add(c);  
}
```

```
void removeListener(Cartella c){  
    cartelle.remove(c);  
}
```

```
private void notifyAllListeners(int x){  
    Iterator<Cartella> iter=cartelle.iterator();  
    while (iter.hasNext()) {  
        iter.next().checkNumber(x);  
    }  
}
```

Banco

```
public int getNextNumber() {
    if (sacchetto.size()==0) {
        System.out.println("NUMERI FINITI!");
        System.exit(1);
    }
    int index=Common.generatore.nextInt(sacchetto.size());
    Integer num=(Integer)sacchetto.get(index);
    sacchetto.remove(index);
    System.out.println("====> ESTRATTO: "+num );
    notifyAllListeners(num.intValue());
    return num.intValue();
}
}
```

Tombola

```
package tombola;
```

```
public class Tombola {
```

```
    public Tombola() {
```

```
        Banco banco = new Banco();
```

```
        Giocatore g1 = new Giocatore("Pippo");
```

```
        banco.addListener(new Cartella(g1));
```

```
        banco.addListener(new Cartella(g1));
```

```
        banco.addListener(new Cartella(g1));
```

```
        Giocatore g2 = new Giocatore("Pluto");
```

```
        banco.addListener(new Cartella(g2));
```

```
    while (true) {
```

```
        int x = banco.getNextNumber();
```

```
        System.out.println("Il numero estratto é + x);
```

```
    }
```

```
}
```

```
public static void main(String[] args) {
```

```
    Tombola x=new Tombola();
```

```
}
```

```
}
```



*Eliminati i riferimenti
Diretti alle cartelle*

run!

Cartella 1 per giocatore Pippo : 5 3 6

Cartella 2 per giocatore Pippo : 8 7 10

Cartella 3 per giocatore Pippo : 1 2 4

Cartella 4 per giocatore Pluto : 1 5 4

====> ESTRATTO: 7

Pippo ha il numero 7 in cartella 2

Il numero estratto é 7

====> ESTRATTO: 5

Pippo ha il numero 5 in cartella 1

Pluto ha il numero 5 in cartella 3

Il numero estratto é 5

====> ESTRATTO: 9

Il numero estratto é 9

====> ESTRATTO: 6

Pippo ha il numero 6 in cartella 1

Il numero estratto é 6

====> ESTRATTO: 3

Pippo ha il numero 3 in cartella 1

Pippo ha vinto con cartella 1

3 5 6