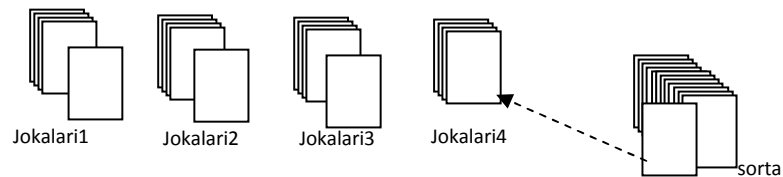


2. Karta jokaldia (2,5 puntu)

Lau lagunek SEIKOEN jokaldia egin nahi dute. Hauek dira jokoaren ezaugarri nagusiak:

1. Karta sortak 40 karta ditu, lau sailetan banatuta (urreak, kopak, ezpatak eta bastoiak). Sail bakoitzak 10 karta ditu, 1etik 10era.
2. Kartak lau jokalarien artean banatuko dira, hau da, jokalaria bakoitzak 10 karta izango ditu.



3. Partida honela antolatuko da:
 - Lehen jokalaria bere lehen karta hartuko du (gainekoa),
 - Ahal badu (ikus behean), mahaian jarriko du.
 - Ezin badu, orduan karta hori bere karta guztien azpian jarriko du
 - Eta hurrengo jokalaria pasako zaio txanda.
4. Jokalari batek ondoko kasuetan jar dezake karta mahaian:
 - Karta seiko bat da.
 - Karta ez da seikoa, baina mahaian badagoen karta baten ondorengoa da. Adibidez, nire karta urrezko bikoa baldin bada, eta mahaian urrezko hirukoa balego, edo bestela nire karta urrezko zortzikoa balitz eta mahaian urrezko zazpikoa balego.
- Jokoa jokalaria batek bere azken karta jartzen duenean amaituko da

Karta_Sorta eta Bilara DMAak erabilia (ez dira inplementatu behar DMA hauen eragiketak) ondokoa eskatzen da:

- a) Problema hau ebazteko erabiliko dituzun datu-egiturak definitu (jokalariak eta mahaiaren egoera adierazteko).
- b) *Diseinatu eta idatzi* azpiprograma bat partida bat simulatu eta irabazlea zein den esango diguna.

KartaSorta DMA:

```
public class Karta {  
  
    String palo;    // urrea, kopa, ezpata, bastoia  
    int balioa;     // 1 eta 10 arteko balioa  
}  
  
public class KartaSorta {  
  
    private Karta[] kartak;  
  
    public KartaSorta () { // eraikitzailea  
        // post: 40 karta daude ausaz  
  
        public Iterator<Karta> iteradore()  
            // kartak aztertzeke iteradorea bueltatzen du  
  
    }  
}
```

Bilara:

```
public class Bilara<T> {  
  
    public Bilara(); // eraikitzailea  
    // bilara hasieratzen du (hutsa)  
  
    public boolean hutsaDa();  
    // true B hutsa baldin bada eta false bestela.  
  
    public void txertatuEzk(T elemento);  
    // E elementua ezkerretik gehitu dio  
  
    public void txertatuEsk(T elemento);  
    // E elementua eskuinetik gehitu dio  
  
    public void ezabatuEzk();  
    // ezkerrean dagoen elementua ezabatu du  
  
    public void ezabatuEsk();  
    // eskuinean dagoen elementua ezabatu du  
  
    public T lortuEzk();  
    // ezkerrean dagoen elementua bueltatzen du  
  
    public T lortuEsk();  
    // eskuinean dagoen elementua bueltatzen du  
  
}
```