

Izen abizenak

HIRU karta-jokoa

HIRU karta-jokoa inplementatu nahi da. Joko honek, antzekotasun ugari ditu UNO edo txupate dos bezalako jokoeekin.

HIRU-ra jolasteko erabiltzen diren kartak honakoak dira: karta arruntak, non hauen ezaugarriak, kolore (String) eta zenbaki (int) diren, eta, salto komodinak, lapurtu komodinak nahiz kolore aldaketa komodina.

Partida jolastuko duten jokalarien zerrenda izango dugu. Jokalari bakoitzaren ezaugarriak, ezizen edo nick (String) eta jolasteko egokitu zaizkion karten zerrenda (ListaKartak) izango dira.

HIRU jokoaren erregelak, ondorengoak dira. Partida baten hasieran, karta batzuk jokalarien artean banatzen dira, eta karta bat mahai erdian jartzen da denok ikusgarri izateko. Gainontzeko kartak piloa ipintzen dira beherantz begira, partidan zehar kartak lapurtzeko erabilgarri izateko. Txanda jarraituz, jokalaria bakoitzak, posible badu behintzat, tokatu zaizkion karten artean bat botako du, mahai erdian dauden karten kopurua handituz¹. Jokalariak ezin du edozein karta bota, erregela batzuk daude eta karta arruntak edo komodinen artean aukeratu beharko du. Karta arruntak, mahai erdian dagoen azken kartaren kolorearekin edo zenbakiarekin bat datorren bat izan behar du (adibidez, mahai gainean utzi den azkeneko karta 5 berde bat izanez gero jokalariai beste 5 bat edo berdea den karta bat bota ahaliko du). Horrelako kartarik izan ezean, komodin bat botatzea posible da, hauek edozein egoeratan bota daitezkeelako. Komodinik ez balu izango orduan txanda-pasa egin beharko du, kartarik bota gabe. Beraz, txanda bakoitzean, gehienez karta bat utzi ahaliko da mahai gainean. Gauzak horrela, irabazlea, karta guztiak botatzea lortzen duen lehenengo jokalaria izango da, baina partida ez da amaituko jokalaria guztiak kartarik gabe gelditu arte, jokalarien artean ranking bat osatzen baita.

Mahaia klaseak partida jolastuko den lekua adierazten du eta beste atributu batzuen artean, txanda (int - txandak kontrolatu ahal izateko, zenbaki honen balioak 1etik jokalariai kopuru arteko tartekoa izango da) eta jolasten ari den egungo kolorea (String) eta zenbakia (int) izango ditu. Orokorrean, jolasten den egungo kolorea eta zenbakia atributuak, mahaiaren erdian ikusgarri dagoen kartarekin (piloko azkena) bat etorriko dira, baldin eta azkena jaurti den karta kolorea aldatzen duen komodia izan ez bada. Kasu konkretu hauek, aurrerago azalduko dira.

Bukatzeko, lapurtzeko dauden karten piloa eta jokalariaiek bota dituzten karten piloa egongo dira (ListaLapurtzekoKartak eta ListaBotatakoKartak klasekoak, ohartu bigarren zerrenda honen azken elementuak ikusgarri dagoen karta irudikatzen duela, hau da, jaurtia izan den azkena).

Ondoren, Jokalari klasearen *public int jolastuKarta()* metodoaren inplementazioa ikus daiteke:

```
public int jolastuKarta(){
    Karta kart;
    int emaitza=1;// paso egiten badu emaitzak 1 balioa jasoko du
    kart=this.kartaZerrenda.hautatuKartaBat(); //dituen karten artean bat hautatuko du

    if(kart!=null)
    {
        //karta hori jokalariairen kartetatik kenduko da eta mahaiaren erdiko piloa utzi
        emaitza=0;
        this.kartaZerrenda.ezabatuKartaListatik(kart);
        ListaBotatakoKartak.getNireListaBotatakoKartak().utzi(kart);
    }
}
```

¹ Partida hasieran ez dago kartarik mahai erdian, partida aurrera joan ahala mahai erdiko karten kopurua handitzen joango da

```

    kart.egikarituKarta();
}
return emaitza;
}

```

- *hautatuKartaBat()* metodo honek, jokalaria dituen karten artetik, posible bada behintzat bat hautatuko du:
 - Lehenengoz inplementatu behar ez duzuen *private Karta eskatuKarta()* erabiliz, erabiltzaileari teklatutik karta bat hautatzeko eskatzen zaio.
 - Teklatutik aukeratutako karta hori jolasterik posible ez bada, *null* balioa itzuliko du *hautatuKartaBat()* metodoak. Kontutan hartu, karta bat jolastu ahal izateko, azken hau komodina edo mahaiko kolore edo zenbakiarekin bat datorren karta arrunt bat izan behar dela.
- *egikarituKarta()* metodo honen helburua, kartaren arabera izango da. Karta arrunta denean, mahaia gordetzen duen kolorea eta zenbakia aldatuko du, berea ezarriz. Komodinen kasuan aldiz, aukera desberdinak gerta daitezke:
 - Salto komodina: bere izenak adierazten duen bezala, txanda bat pasa arazten du, hau da, hurrengo jokalaria txanda pasa egin arazi egiten dio, txanda, hurrengoaren hurrengo jokalaria pasaz. *public void txandaPasa()* Mahaia inplementatuta duen metodo bat izango da (zuek ez duzue inplementatu behar bakarrik erabili).
 - Lapurtu2 komodina: Egingo duen lehenengo gauza hurrengo jokalaria zein den identifikatzea izango da eta bi karta lapurtzea behartuko dio *public void lapurtu2Karta()* metodoaren bitartez. Azken metodo hau, Jokalaria klasean egongo den metodo bat izango da. Bere helburua, Lapurtzeko pilotik amaieran dauden bi karta zerrendatik ezabatu eta jokalaria karten pilora gehitzeaz arduratuko da.
 - Kolore aldaketa komodina: kolore atributu batez bereizten da. Honela, komodina botatzean Mahaiko kolorea bere kolorez aldatuko du.

Bukatzeko, jolastutako partiden ranking-en zerrenda bat dugu. Bertan, partida bakoitzeko jokalarien ranking-a soilik gordetzen da, beti ere modu ordenatuan, irabazlea lehenengoa izanik, bigarrena bigarren etab.

Eskatzen dena:

- **(0.5 puntu)** Klase diagrama marraztu, adieraziz klase bakoitzeko zeintzuk diren atributuak eta metodoak. Eta zeintzuk diren ere, klaseen arteko erlazioak. Diseinuak informazioaren partekaketa maximizatzea lortu behar du.
- **(0.4 puntu)** *ListaPartidak* klaseko *public int txapeldunKopurua(Jokalaria pJokalaria)* metodoaren sekuentzia diagrama irudikatu. Metodo honek, *pJokalaria* zenbat partidetan geratu den lehenengo hiru posturen artean itzuliko du.
- **(0.1 puntu)** Inplementatu *Karta* klase eta gainontzeko azpiklaseen eraikitzailea. Baita dauden EMA klaseenak ere.
- **(0.5 puntu)** Inplementatu *egikarituKarta()* eta *hautatuKartaBat()* behar den klase guztietan.

OHARRAK:

Penalizazioa jasoko duzue baldin eta:

- Beharrezkoak ez diren metodoen inplementazioak, nahiz eta beste testuinguru batean beharrezkoak izan.

- Eskatzen diren metodo batzuk, ez dira zertan beste klase batetik ikusgarriak izan behar. Penalizazioa jasoko duzue baldin eta ikusgarria izan behar ez duen metodo bat ikusgarri definituz gero.
- Orain arte ikusi eta erabilitako arauak ez jarraitzea, alegia, elementu desberdinen izenak hautatzeko arauak (atributuak, metodoak, parametroak, aldagaiak).