

# PROGRAMAZIO MODULARRA ETA OBJEKTUETARA BIDERATUTAKO PROGRAMAZIOA

Hirugarren Azterketa Partziala // Azterketa finalaren lehen zatia (maiatzaren 28)

## 1. Ariketa - Ekitaldien antolaketa (1.5 puntu)

Bizkaiko Oyuken EM enpresak, mota guztietako ekitaldiak antolatzen ditu. Berauen bezero bakoitzeko, datu batzuk gordetzen dira, besteak beste, zerga identifikazio kodea ( IFK ), izena eta kontratatutako ekitaldien zerrenda. Zerrenda hau, ekitaldien burutze dataren arabera ordenaturik egongo da.

Bezeroez gain, Oyuken EM enpresak, bertan lan egiten duten langileen informazioa gordetzen du zerrenda betean. Langile bakoitzeko, NAN zenbakia, izen osoa eta ariketa honetarako beharrezkoak ez diren beste zenbait datu gordetzen dira.

Ekitaldi mota ezberdinak bereizten dira, hala nola, zeremoniak, festak, jaialdiak eta hitzaldiak. Ekitaldi guzti hauek, alfanumeriko izango den identifikadore batez, gertaera burutuko den lekuaz eta hasiera dataz (Date motakoa) bereizten dira.

Zeremoniak eta festak, egun bakar batean ospatzen dira, eta ezaugarri bezala, deskribapen txiki bat (adibidez "Bilboko Petanka Txapelketaren Sari Banaketa") eta arduradun bat izango dute (Langile motakoa). Horrez gain, zeremonia ekitaldi kasu zehatzerako, zeremonia mota gordetzen da (adibidez, "ezkontza"). Festarako aldiz, ekitaldiaren iraupena (ordutan) gordetzen da.

Jaialdiak eta hitzaldiak, hainbat egun jarraietan ospatzen direnez, ekitaldi batek dituen beste ezaugarriez gain, amaiera data (Date motakoa) ere izango dute ezaugarritzat. Jaialdiek, gainontzeko atributuez gain, logotipo (Image motakoa) bat izango dute ezaugarri gisa. Hitzaldiek, aurrez aipaturiko atributu guztiez gain, edizioa gertaera zenbakia eta maiztasuna izango dute ezaugarritzat (adibidez, "urtero").

Horrez gain, jaialdi nahiz hitzaldietan, egun bakarreko ekitaldietan ez bezala, ez da inongo arduradunik egongo. Bere ordean, egun bat baino handiagoko ekitaldi bakoitzerako, zereginen lista bat gordeko da, non zeregin bakoitzak arduradun bat izango duen (Langile motakoa). Zeregin bakoitza, deskribapen txiki batez (adibidez, "Hitzaldi aretoaren apainketa") eta iraupen ordu kopuru batez ezaugarritzen dira.

**OHARRA:** Ebaluazio finala aukeratu duzuenok, aurreneko zati honetako 1 eta 2 atalak soilik burutu beharko dituzue, izan ere, ondoren, bigarren proba bat burutu beharko baituzue.

Burutu beharrekoa:

1. (0,60 puntu) Marraztu ezazu klase diagrama, atributuak eta metodoak adieraziz, berauen ikusgarritasunarekin batera (private -, public +, protected ▲). Klaseen arteko erlazioak ere adieraz itzazu.

2. **(0,40 puntu)** Marraztu ezazu `ListaZereginak` klaseko honako burukoa duen `public int zenbatOrdu(Langile pLangile)` metodoaren **sekuentzia diagrama**. Metodo honek, `pLangile` arduradunak dituen zereginei eskainitako ordu kopuru totala itzuliko du.
3. **(0,10 puntu)** Ekitaldia klasearen eta honekin herentzia harreman duten gainontzeko klaseen **eraikitzaileak** inplementa itzazu.
4. **(0,40 puntu)** `Bezeroa` klaseko `public int zenbatOrdu(Langile pLangile)` **metodoa** inplementa ezazu. Metodo honek beste klaseetako metodoren bat edo klase bereko metodoren bat erabili ezker, inplementa itzazu hauek ere (metodoak zein klasetakoak diren argi azalduz). Metodo honen helburua, `bezeroak kontratatuak` dituen `ekitaldien zerrenda` zeharkatu eta `arduradun gisa pLangile` duten egun bat baino gehiagoko `ekitaldien zereginei` dedikatutako ordu kopurua itzultzea izango da.

**OHARRA:** Penalizazioa jasoko duzue baldin eta:

- Soilik, eskatutakoa garatzeko beharrezkoak diren metodoak definitu eta inplementa itzazu, ez bat gehiago bat gutxiago.
- Eskatzen diren metodo batzuk, ez dira zertan beste klase batetik ikusgarriak izan behar. Penalizazioa jasoko duzue adibidez baldin eta ikusgarria izan behar ez duen metodo bat ikusgarri definituz gero etab.
- Penalizazioa jasoko duzue baita baldin eta ondorengo elementuei ez bazaie klasean ikusitako normak aplikatzen: atributuei, metodoei, parametroei, aldagaiei ...
- Atributuak atzitzeko, `getter/setter` metodoak erabili beharko dira. Txartzat emango da baldin eta atributuak zuzenean atzitzen badira.

## **2. Ariketa - Labirinto itsua TeEme (1 puntu)**

*Itsu Maze TM*, bi jokalaria edo gehiagorentzako, estrategia eta zoria bateratzen dituen joko bat da. Jokoa, ohol “itsu” baten gainean jarduten da, hemendik bere izena, jokalariek inoiz ez baitute jakingo oholeko zein laukitxoetan dauden kokatuta. Jokoaren hasieran, jokalaria guztiak hausaz kokatuko dira oholeko laukitxo batean eta ondoren txandaka parte hartuko dute. Aurrerago azalduko den bezala, txanda dagokion jokalaria, oholean behin edo gehiagotan mugitzeko aukera izango du txanda berean. Jokoaren helburua, bizirik irautea da, eta jokalaria bakarria geratzen denean amaituko da, berau izanik irabazlea.

- Mugimendu posibleak, jokalaria kokatuta dagoen ondoko laukitxoetara izango dira, hau da, aurrera ('i'), atzera ('h'), eskumara ('e') eta ezkerretara ('m').
- Jokalaria, hutsik dagoen laukitxo batera mugitzen bada, berriz mugitu beharko da. Beraz, egoera hau den bitartean bere txanda ez da agortuko eta mugitzen jarraitu beharko du.

- Jokalaria, beste jokalaria bat dagoen laukitxo batetara mugituz gero, bertan zegoen jokalaria kanporatua izango da! Jokalariaren txanda agortu egingo litzateke.
- Jokalaria, oholeko kokaguneetatik irteten bada (aurreneko lerroan egonik aurrera mugitzen bada, edo azkeneko zutabean egonik eskumatara mugitzen bada, etab), berriz ere hausaz libre dagoen oholeko laukitxo batean kokatuko da, noski, beti ere kokagune ezezaguna izaten jarraituko du jokalariaientzat eta txanda agortu egingo litzaioke.

*Itsu Maze TM* jokoa implementatzeko klase diagrama bat diseinatu da, Teklatua (EMA), Koordinatua (DMA), Ohola (EMA) eta Jokalaria (DMA) izanik ariketa honetarako interesa duten klaseak:

```
public class Teklatua
{
    ...
    public static Teklatua getTeklatua()
        // Irteera: Teklatuaren instantzia bakarra itzultzen du (Singleton patroia)
    { ... }

    public char hizkiaIrakurri()
        // Irteera: Erabiltzaileak teklatutik sartutako hizkia itzultzen du.
        // Eragina: Ez du salbuespenik propagatzen, guztiak barnean tratatzen ditu
    { ... }
}
```

```
public class Koordinatua
{
    ...
    public void atxikitu(char pNorantza) throws NorantzaOkerraException
        //Eragina: Koordinatua aldatzen du. Koordinatu berria, aurreko posizioa kontuan izanda
        //erabiltzaileak sartutako pNorantza arabera izango da, 'i' aurrera, 'h' atzera,
        //'e'eskumara eta 'm' ezkerretara. Salbuespena jaurtiko du baldin eta erabiltzaileak
        //sartutako hizkia ez bada aurrez aipatutako bat.
    { ... }
}
```

```
public class Ohola
{
    ...
    public static Ohola getOhola()
        // Irteera: Oholaren instantzia bakarra itzultzen du (Singleton patroia)
    { ... }

    private Koordinatua koordinatuaJokalaria(Jokalaria pJokalaria)
        // Aurrebaldintza: pJokalaria oholean egongo da.
        // Irteera: pJokalariaren oholeko koordinada itzuliko du.
    { ... }

    private void gehituJokalaria(Jokalaria pJokalaria, Koordinatua pKoord) throws
LaukitxoaOkupatutaException, IndexOutOfBoundsException
        // Eragina: pJokalaria oholeko pKoord posizioan gehituko du.
        //IndexOutOfBoundsException jaurtiko du baldin eta pKoord ez bada baliozko
        //posizio bat. pKoord posizioan jokalaria bat badago, hau jokotik kanporatuko da
        // eta LaukitxoaOkupatutaException salbuespena jaurtiko du.
    { ... }

    public void gehituJokalariaLaukitxoanAusaz(Jokalaria pJokalaria)
        // Eragina: pJokalaria, hausaz, oholeko hutsik edo libre dagoen posizio batean
        //gehituko du.
    { ... }

    public void jokalariaOholetikEzabatu(Jokalaria pJokalaria)
        // Eragina: pJokalaria oholetik ezabatuko du.
    { ... }

    public void mugitu (Jokalaria pJokalaria, char pNorantza) throws NorantzaOkerraException,
OholKokagunetatiKanpoException, LaukitxoaOkupatutaException
```

```

        // Eragina: pJokalaria pNorantza norabidean mugituko du. NorantzaOkerraException
        //salbuespena propagatuko du baldin eta pNorantza ez bada 'i', 'h', 'e' edo 'm'.
        // OholKokagunetatiKanpoException albuespena jaurtiko du baldin eta mugimendua
        // ezin bada bete, oholetik kanpora baitoa. LaukitxoaOkupatutaException albuespena
        //propagatuko du baldin eta mugimenduarekin jokalaria bat ezabatu bada.

        {
            koordenatua koord = this.koordenatuaJokalaria(pJokalari);
            koord.atxikitu(pNorantza);
            this.jokalariaOholetikEzabatu(pJokalari);
            this.gehituJokalaria(pJokalari, koord);

            //TODO implementa ezazu metodo hau.
        }
    }
}

```

```

public class Jokalaria
{
    ...

    public void mugimenduaBurutu() throws SaiakerakAmitutaException
    {
        // Eragina: Jokalariaren mugimendua burutuko du bere txanda denean.
        // SaiakerakAmitutaException albuespena jaurtiko du baldin eta 3 saiakeren
        // ondoren jokalaria ez badu norantza hizkia ('i', 'h', 'e' edo 'm') ongi ipini.

        boolean txandaAmitu = false;
        while (!txandaAmitu){

            //TODO implementa ezazu metodo hau.

        }
    }
}

```

Burutu beharrekoa:

1. **(0,40 puntu)** Ohola klaseko *mugitu(Jokalaria pJokalari, char pNorantza)* metodoa implementa ezazu.
2. **(0,55 puntu)** Jokalaria klaseko *mugimenduaBurutu()* metodoa implementa ezazu.
3. **(0,05 puntu)** NorantzaOkerraException eta LaukitxoaOkupatutaException klaseak implementa itzazu.

**OHARRA:** Soilik aurreko puntuetan eskatzen diren metodoak implementa itzazu. Eskaintzen diren gainontzeko metodoak erabil ditzakezue, baina ez dira zertan implementatu behar.