

|                    |                       |
|--------------------|-----------------------|
| <b>Ikasturtea:</b> | 2018/2019             |
| <b>Gaia:</b>       | Software Ingeniaritza |
| <b>Ariketak:</b>   | Patroiak              |
| <b>Data:</b>       | 2019ko Otsaila        |

### 1. Ariketa

Bezzeroaren eskaerak tratatzen dituen aplikazio informatiko bat dugu martxan. Aplikazio horretan bezzeroaren izena eta deitura String motako objektu bakarrean biltegiratzen dira. Oraintsu, software berri bat erosi dugu gure lehengo aplikazio zaharrarekin funtzionatu behar duena baina software berri honetan bezzeroaren izena eta deitura bi String desberdinetan biltegiratzen dira.

Bi aplikazioak birprogramatu baino lehen eta hori ekiditeko, horrek suposatzen duen lanarekin programazio aldetik, ebazpen “garbiagoa” aurkitu nahi da eta diseinu patroiak erabili nahi dira arazoari irtenbidea emateko.

- Aipatu egoera honetan kontutan hartu beharreko SOLID printzipio bat eta arrazoitu zergatik den garrantzitsua.
- Labur arrazoitu zein patroi erabiliko zenituzkeen eskatutakoa inplementatzeko.
- Klase diagrama zehaztu klase nagusiekin eta garrantzitsuak iruditzen zaizkizun metodoekin.
- Klaseen inplementazioa idatzi.

### 2. Ariketa

Zenbaki multzo bat gordetzen duen klase bat dugu. Zenbaki hauekin operazio multzo bat erabili nahi dugu (batura, batazbestekoa...). Baina ez dugu klasearen inplementazioa operazioekin kutsatu nahi, beste testuinguru batzuetan berrerabili daiteke eta. Horregatik, operazioak beste klase batetan bildu ditugu. Honetaz gain, zenbaki bat gehitu/ezabatzen denean, operazio guztiak birkalkulatzeari nahi dugu.

- Aipatu egoera honetan kontutan hartu beharreko SOLID printzipio bat eta arrazoitu zergatik den garrantzitsua.
- Labur arrazoitu zein patroi erabiliko zenituzkeen eskatutakoa inplementatzeko.
- Klase diagrama zehaztu klase nagusiekin eta garrantzitsuak iruditzen zaizkizun metodoekin.
- Klaseen inplementazioa idatzi (eskeletoa, metodoen gorputza gabe).

### 3. Ariketa

Duela gutxi hiru enpresa desberdin fusionatu dira. Egin nahi dena:

- Hiru enpresak eta bere klase hierarkia banatuta mantendu.
- Hiru enpresen fusioan, lehenengo eta bigarren enpresek metodo berdinak dituzte (*Langileak.getLangileak()*). Hirugarrenak aldiz, atzeritarra denez, bere metodoak ingelesez ditu (adibidez, *Employment.getEmployees()*). Enpresa sisteman gehitu nahi dugu jada martxan dauden sistemak ukitu gabe.
- Langileen sorrera era orokor batetan kudeatu (*langileaSortu(int empresa, Langilea lan)*)
- Enpresaren ikuspegi orokor bat eman, enpresa bakarra izango balitz bezala.

Eskatzen da:

|                    |                       |
|--------------------|-----------------------|
| <b>Ikasturtea:</b> | 2018/2019             |
| <b>Gaia:</b>       | Software Ingeniaritza |
| <b>Ariketak:</b>   | Patroiak              |
| <b>Data:</b>       | 2019ko Otsaila        |

- Aipatu egoera honetan kontutan hartu beharreko SOLID printzipio bat eta arrazoitu zergatik den garrantzitsua.
- Labur arrazoitu zein patroi erabiliko zenituzkeen eskatutakoa implementatzeko.
- Klase diagrama zehaztu klase nagusiekin eta garrantzitsuak iruditzen zaizkizun metodoekin.
- Klaseen inplementazioa idatzi.

#### 4. Ariketa

Kalitate irizpideek enpresa sailetan antolatzera eramanez. Ondorioz, enpresa bakoitza sail desberdinekin osatuko dute. Era berean sail bakoitzak langileak edo beste sail batzuk izan ditzake. Egitura honetan egin nahi diren operazioak:

- Sail bakoitzean dauden langileak bistaratu.
- Enpresako langile zenbakia bueltatu.

Eskatzen da:

- Aipatu egoera honetan kontutan hartu beharreko SOLID printzipio bat eta arrazoitu zergatik den garrantzitsua.
- Labur arrazoitu zein patroi erabiliko zenituzkeen eskatutakoa implementatzeko.
- Klase diagrama zehaztu klase nagusiekin eta garrantzitsuak iruditzen zaizkizun metodoekin.
- Klaseen inplementazioa idatzi.

#### 5. Ariketa

Enpresa bateko langileek kategoria bat dute (1,2,3...N). Soldata igotzeko momentuan erabiltzen diren ebaluazio irizpideak kategoriaren arabera dira, kategoria desberdinekin ebaluazio berdina izan dezaketelarik. Soldata igoera hau baloratzeko Soldata izeneko klase bat sortzea erabaki da, *soldata.go?* metodo estatikoarekin. Metodo honek langile konkretu batek irizpideak betetzen dituen edo ez konprobatuko du.

Eskatzen da:

- Aipatu egoera honetan kontutan hartu beharreko SOLID printzipio bat eta arrazoitu zergatik den garrantzitsua.
- Labur arrazoitu zein patroi erabiliko zenituzkeen eskatutakoa implementatzeko.
- Klase diagrama zehaztu klase nagusiekin eta garrantzitsuak iruditzen zaizkizun metodoekin.
- Klaseen inplementazioa idatzi.

|                    |                       |
|--------------------|-----------------------|
| <b>Ikasturtea:</b> | 2018/2019             |
| <b>Gaia:</b>       | Software Ingeniaritza |
| <b>Ariketak:</b>   | Patroiak              |
| <b>Data:</b>       | 2019ko Otsaila        |

## 6. Ariketa

Software Ingeniaritzako praktika inplementatzeko garaian, hurrengo klasea idatzi dute ikasle batzuek:

```
public class DBKudeatzailea {

    private static DBKudeatzailea INSTANCE = null;
    private static Connection konexioa;

    private DBKudeatzailea () {
        try{

            Class.forName("driverra");
            konexioa = DriverManager.getConnection("helbidea");
            konexioa.setAutoCommit(false);

        }
        catch(Exception e){
            System.out.println(e.toString());
        }
    }

    public static DBKudeatzailea getInstance() {
        if (INSTANCE == null) {
            INSTANCE = new DBKudeatzailea();
        }
        return INSTANCE;
    }

    public void konexioaIreki() {...}
    public void konexioaItxi() {...}
    public Bezeroa getBezeroa (String id) {...}
    ...
}
```

- Zein patroi erabiltzen ari dira?
- Zergatik aukeratu dutela iruditzen zaizu?

|                    |                       |
|--------------------|-----------------------|
| <b>Ikasturtea:</b> | 2018/2019             |
| <b>Gaia:</b>       | Software Ingeniaritza |
| <b>Ariketak:</b>   | Patroiak              |
| <b>Data:</b>       | 2019ko Otsaila        |

## 7. Ariketa

Telefono konpainia batek zerbitzu berri bat eskaini die bere bezeroei. “Dei abisua” zerbitzuak bezeroak aukeratutako gailuetara (posta elektronikoa, bilagailua, mugikorra edo telefono finkoa) mezu bat bidaltzen du bere gailuetako batetan dei bat jaso dela abisatuz. Mezuan deia jaso duen gailuaren zenbakia azalduko da. Dei bezala gailu batek egin dezakeen edozein kontaktu ulertuko dugu.

Gailu bakoitzean (adibidez *gailu1*) deia jaso denean beste gailu batetan (adibidez *gailu2*) mezua jaso nahi dugula adierazteko *gailu1.harpidetu(gailu2)* metodoa erabiliko dugu eta jada mezu gehiago jaso nahi ez dugunean *gailu1.harpidetzaEten(gailu2)* metodoa.

Programa nagusia eta bere irteera ondokoak dira:

```
public static void main (String[] args) {
    TelefonoFinko tfinko = new TelefonoFinko ("943015052");
    TelefonoMugikor tmugikor = new TelefonoMugikor ("Nire Mugikorra");
    PostaElektronikoa posta = new PostaElektronikoa ("si@ehu.es");
    Bilagailua bi = new Bilagailua ("Nire Bilagailua");

    tfinko.harpidetu(tmugikor);
    tfinko.harpidetu(bi);
    tfinko.deia(new TelefonoFinko("943015555"));
    tfinko.harpidetzaEten(bi);
    tfinko.deia(posta);

    posta.harpidetu(tmugikor);
    posta.harpidetu(tfinko);
    posta.deia(bi);
}
```

```
Mezua bilagailura, 943015052-n 943015555-k egindako deia jaso da.
Mezua mugikorrera, 943015052-n 943015555-k egindako deia jaso da.
Mezua mugikorrera, 943015052-n si@ehu.es-k egindako deia jaso da.
Mezua finkora, si@ehu.es-n Nire bilagailua-k egindako deia jaso da.
Mezua mugikorrera, si@ehu.es-n Nire bilagailua-k egindako deia jaso da.
```

- Aipatu egoera honetan kontutan hartu beharreko SOLID printzipio bat eta arrazoitu zergatik den garrantzitsua.
- Labur arrazoitu zein patroi erabiliko zenituzkeen eskatutakoa implementatzeko.
- Klase diagrama zehaztu klase nagusiekin eta garrantzitsuak iruditzen zaizkizun metodoekin.
- Klaseen implementazioa idatzi (eskeletoa, metodoen gorputza gabe).

|                    |                       |
|--------------------|-----------------------|
| <b>Ikasturtea:</b> | 2018/2019             |
| <b>Gaia:</b>       | Software Ingeniaritza |
| <b>Ariketak:</b>   | Patroiak              |
| <b>Data:</b>       | 2019ko Otsaila        |

## 8. Ariketa

Dokumentuak idazteko editore bat sortzea eskatzen digute. Dokumentu bakoitzak orrialdeak ditu, orrialde bakoitzak testu lerroz osatutako zutabeak dituelarik. Testu lerroak karakterez osatuta daude.

Zutabe eta orrialdeek markoak izan ditzakete eta markoek barruan zutabeak izan ditzakete.

Zutabe, marko eta testu lerroek irudiak izan ditzakete.

- Aipatu egoera honetan kontutan hartu beharreko SOLID printzipio bat eta arrazoitu zergatik den garrantzitsua.
- Labur arrazoitu zein patroi erabiliko zenituzkeen eskatutakoa implementatzeko.
- Klase diagrama zehaztu klase nagusiekin eta garrantzitsuak iruditzen zaizkizun metodoekin.
- Klaseen inplementazioa idatzi (eskeletoa, metodoen gorputza gabe).